VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Kontextová analýza textu

Context-based Text Analysis

2018 Krištof Kubík

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Krištof Kubík**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Kontextová analýza textu
Context-based Text Analysis

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je zmapovat techniky používané pro automatickou analýzu textu z pohledu detekce významu slov v rámci jejich kontextu. Cílem je implementovat některé ze zjištěných metod.

Práce bude obsahovat:
1. State of the art.
2. Podrobný popis zvoleného/zvolených algoritmů.
3. Experimenty a jejich vyhodnocení (možno použít tabulky a grafy).
4. Závěr - zhodnocení výsledků.

Seznam doporučené odborné literatury:

[1] Aggarwal, Charu C., and ChengXiang Zhai, eds. Mining text data. Springer Science & Business Media, 2012.
[2] Berry, Michael W. "Survey of text mining." Computing Reviews 45.9 (2004): 548.
[3] Cohen, Aaron M., and William R. Hersh. "A survey of current work in biomedical text mining." Briefings in bioinformatics 6.1 (2005): 57-71.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jan Platoš, Ph.D.**

Datum zadání: 01.09.2017
Datum odevzdání: 13.07.2018

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave, dňa 12. 7. 2018 .....

Rád by som na tomto mieste poďakoval predovšetkým doc. Ing. Janovi Platošovi, Ph.D za odborné konzultácie a vedenie mojej diplomovej práce

Abstrakt

Práca študuje aktuálne metódy pre kontextovú analýzu textu. Nasleduje zoznámenie sa s problematikou a predstavuje aktuálne trendy. Práca uvádza základné teoretické pojmy a metódy celého procesu analýzy. Opisuje sa tu zdroj dát pre slovenský a anglický jazyk. Na základe teoretický poznatkov sú implementované zvolené metódy. Na experimentoch je dokázaná ich úspešnosť a porovnané ich vlastnosti.

Kľúčové slová: kontext, text, analýza, korpus, N-gram

Abstract

Diploma thesis studies current methods for context-based text analysis. It follows familiarization with the issues and presents current trends. The thesis presents basic theoretical concepts and methods of the whole analysis process. Describes a source of data for the Slovak and English languages. On the basis of the theoretical knowledge, the chosen methods were implemented. The experiments demonstrate their success and compare their properties.

Key Words: context, text, analysis, corpus, N-gram

Obsah

Zoznam obrázkov **8**

Zoznam tabuliek **9**

1 Úvod **10**

2 Súčasný stav problematiky **11**
2.1 Dolovanie informácií z textu . . . 11
2.2 Textové dokumenty . . . 11
2.3 Kontextová analýza . . . 12
2.4 Druhy analýz . . . 12
2.4.1 Štatistický prístup k analýze textu . . . 12
2.4.2 Lingvistický prístup k analýze textu . . . 13
2.4.3 Kontextové spracovanie textu . . . 15
2.5 Metódy pre analýzu textu . . . 15
2.5.1 Lematizácia a morfologická analýza . . . 15
2.5.2 Eliminácia neplnýznamových slov . . . 15
2.5.3 Váhovanie a normovanie termov . . . 16
2.6 Metódy pre klasifikáciu textov . . . 16
2.6.1 K–najbližších susedov . . . 16
2.6.2 Naivný Bayesov Klasifikátor . . . 17
2.6.3 Rozhodovacie stromy . . . 18
2.6.4 Support Vector Machines . . . 19
2.6.5 Neurónové siete . . . 20

3 Zdroj dát **22**
3.1 Korpus . . . 22
3.2 Slovenský Národný Korpus . . . 23
3.3 Wikipédia . . . 23

4 N-gram **25**
4.1 Algoritmy pre extrakciu N-gramov . . . 25
4.2 Sufixové pole . . . 25
4.3 Rozšírené sufixové pole . . . 26
4.4 Sufixový strom . . . 27
4.5 Algoritmus pre zostrojenie sufixového stromu . . . 28
4.6 Algoritmus invertovaného indexu . . . 31

5 Kategorizácia **32**
5.1 Kategorizácia prostredníctvom extrakcie N-gramov 32
5.2 Predspracovanie textu 33
5.2.1 Konverzia vzstupného textu 34
5.2.2 Členenie slov z textu 35
5.2.3 Eliminovanie neplnovýznamových slov 35
5.2.4 Lematizácia, stemming 36
5.2.5 Lematizácia v angličtine 37
5.2.6 Lematizácia v slovenčine 38
5.3 Proces kategorizácie textu 39
5.4 Porovnávanie profilov 40
5.5 Metódy pre porovnávanie profilov 40
5.5.1 Porovnávanie profilov na základe vzdialeností a pozície N-gramu 40
5.5.2 Porovnávanie profilov na základe váženej početnosti 41

6 Experimenty **42**
6.1 Testovanie a vyhodnotenie 42
6.1.1 Testovanie úspešnosti kategorizácie na základe jazyku 43
6.1.2 Testovanie úspešnosti kategorizácie na základe dĺžky textu 44
6.1.3 Výsledky pre metódu vzdialenosti a pozície N-gramov 44
6.1.4 Výsledky pre metódu váženej početnosti 45
6.2 Porovnanie zvolených metód 46

7 Záver **49**

Literatúra **50**

Zoznam obrázkov

1 Sekvenčný stratifikačný model jazykovej analýzy[2] 14
2 Príklad použitia metódy K–najbližších susedov 17
3 Naivný Bayesov Klasifikátor 18
4 Rozhodovací strom 19
5 Rozhodovacie hranice, optimálna rovina A 20
6 Príklad neurónovej siete 21
7 Príklad taxonómie Wikipédie 24
8 Sufixové pole reťazca mississippi$ ($ značí koniec reťazca) 26
9 Lcp intervalový strom nad reťazcom S = acaaaacatat$ 27
10 Sufixový strom 28
11 Algoritmus pre zostrojenie sufixového stromu 30
12 Zipfianová distribúcia frekvencií N-gramov z technického dokumentu 32
13 Schéma predspracovania textových údajov 34
14 Proces kategorizácie textu 39
15 Meranie vzdialenosti 41
16 Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Slovenský jazyk 44
17 Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Anglický jazyk 45
18 Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Slovenský jazyk 45
19 Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Anglický jazyk 46
20 Porovnanie zvolených metód. Kategória: šport 47
21 Porovnanie zvolených metód. Kategória: politika 47
22 Porovnanie zvolených metód. Kategória: veda 48

Zoznam tabuliek

1 Informácie o vygenerovaných kolekciach SNK 23
2 Informácie o vygenerovaných kolekciach Wikipedia 24
3 Test kategórie šport 43
4 Test kategórie politika 43
5 Test kategórie veda 43

1 Úvod

Žijeme vo svete, kde informácie majú veľkú hodnotu a množstvo takýchto dostupných informácií sa v posledných rokoch výrazne rozrástla. Existuje toľko informácií okolo nás, že sa stáva problém nájsť tie, ktoré sú pre nás dôležité alebo vôbec pravivé. Kvôli tomu existuje veľa databáz a digitálnych encyklopédií rozdelených do kategórií, ktoré pomáhajú používateľom navigovať sa vo svete informácií, ktoré chce získať. Väčšina z týchto informácií sú vo forme textu a práve tu pomáha kontextová analýza textu alebo teda kategorizácia. Kategorizácia textu sa zaoberá problémom výberu kategórie z databázy, ktorá má spoločné znaky s vybratým textom. Obvykle sú tieto databázy vytvárané ľuďmi. Vytvorenie takejto databázy je vysoko časovo náročné, pretože je nutné prečítať každý text (alebo jeho časť), aby mu bolo možné prideliť správnu kategóriu. To je jeden z dôvodov, prečo existuje pomerne veľa výskumov v oblasti automatickej kategorizácie textu. Zvyčajne sa systém na klasifikáciu textu pokúša kategorizovať dokumenty podľa dvoch charakteristík - jazyka a témy konrétneho textu.

Natrénovaný systém sa používa na priradenie kategórie k dosiaľ neznámym (nekategorizovaným) dokumentom. Použité metódy sa líšia od jednoduchých až po pomerne zložité. Zvyčajne je text reprezentovaný ako poradie slov a pre nájdenie správnej kategórie je nutné poznať o takomto texte veľa informácií (počet slov, vzťahy medzi nimi atď.). Z toho vyplývajú aj negatíva ako náročnosť na zdroje a výpočtový čas pričom výsledky nie su vždy uspokojivé.

V tejto práci sa zameriavam na Slovenský a Anglický jazyk. Pre oba jazyky sú dostupné vhodné zdroje dát - korpusy. Cieľom práce je zmapovať metódy pre kategorizáciu textu a vybrané implementovať. Na experimentoch ďalej overiť ich úspešnosť a porovnať ich výsledky.

2 Súčasný stav problematiky

2.1 Dolovanie informácií z textu

Dolovania znalostí z textu v angličtine známe ako Data Mining je jeden z najrýchlejšie sa rozvíjajúcih oborov súčastnosti. Dolovanie textu sa prevádza nad dátovymi zdrojmi ktoré obshajú textové dáta a navyše obsahujú ďalšie potrebné informácie ktoré tieto dáta popisujú. Vytváranie takýchto dát zaberá množstvo času a spotrebúva veľa zdrojov.

V tejto práci sa venujem určitej oblasti dolovania textu ktorá priamo nevyplýva z jej názvu. V mojom prípade sa teda zaoberám o analýzu textových dát, hľadanie kontextu. Zahrnuje to možnosti analýzy napríklad spravodajských článkov alebo deketcie spamu v e-mailových správach. Uplatnenie má tento obor aj v odhaľovaní plagiátorstva.[16]

Pre prácu s textovými dátmi je nutná fáza predspracovania, ktorá eliminuje unikátnosti každého prirodzeného jazyka. Cielom takéhoto spracovania je dokument, ktorý bude dobre čitatelný pre stroje a vhodný na ďalšiu prácu s ním.[1]

2.2 Textové dokumenty

Problémom textových dokumentov je ich samotná reprezentácie a častokrát nemožnosť ich hneď strojovo spracovať. V textových dokumentoch sa s veľkou pravdepodobnosťou môže vyskytovať akékoľvek slovo alebo termín, používaný bežne v prirodzenom jazyku. Ide o tzv. vysokú dimenziu vlastností. Tak isto sa dá predpokladať že v texte sa budú vyskytovať odborné termíny a slová v jazyku odlišnom od jazyka v ktorom je dokument napísaný. Tým pádom je reprezentácia takýchto textových dokumentov veľmi rozsiahla. [1]

Ďalšou črtou je tzv. riedkosť vlastností. Táto vlastnosť dokumentov hovorí o malom počte výskytu dôležitých informácií v dokumente. Inak povedané to znamená že, text môže byť pomerne rozshialy ale nemusí obsahovať dostatočné množstvo slov a termínov z prirodzeného jazyka, na základe ktorých by bolo možné určiť samotné vlastnosti takéhoto dokumentu.

Textové dokumenty sú častokrát neštruktúrované ale i tak z lingvistického poľadu každý dokument obsahuje syntaktickú a sémantickú štruktúru. Z typografického pohľadu sa v dokumentoch nachádza veľa elementov ako sú nadpis, podčiarknutý text, odsadenia, tabuľky, vďaka ktorým je možné určit vlastnosti dokumentu. Dokumenty, ktoré nie sú zaradené do žiadnej kategórie a nie sú dobre štruktúrované, sa volajú slabo štruktúrované. Ide prevažne o novinové články a rôzne bankové správy. Na opačnom konci sa nachádzajú štruktúrované dokumenty ktoré vyššie popísané vlastnosti obsahujú.[1][2]

2.3 Kontextová analýza

Kontextová analýza zisťuje spôsob, akým sa frázy vo vete vzťahujú na objekty reálneho sveta, uvádza vetu do súvislosti s kontextom. Znamená to priradiť frázy jazyka k reálnym objektom vonkajšieho sveta, resp. priradzovať k sebe tie frázy, ktoré sa vzťahujú na jeden a ten istý vonkajší objekt. [3]. Najčastejšie používané sú modely založené na pravdepodobnosti a deterministické modely.

2.4 Druhy analýz

2.4.1 Štatistický prístup k analýze textu

Pri tomto prístupe je kontextová analýza založená na určení identifikátora ktorý popisuje dokument.[2] V štatistickom prístupe analýzu vystihujú nasledovné body:

- Neplnovýznamové slová - tieto slová sa z analýzy bežne vylučujú pretože neposkytujú žiadny význam a teda plnia iba synktatické funkcie. V literatúre sú známe ako stop-slová [3].
- Synonymné výrazy – Slová, výrazy s rovnakým alebo podobným významom (napr. telefón, mobil). Tieto výrazy sa v analýze reprezentujú rovnakým termom. Na odhalenie podobnosti výrazov nepostačuje skúmanie na úrovni morfológie ani syntaxe, potrebná je slovotvorná, sémantická a pragmatická analýza. [2].
- Homonymia – Slová s náhodne totožnou formou a s rôznym významom (napr. kohútik - zviera, kohútik [na vodu]). Takéto slová sa v analýze rozlišujú a rozličními termami. Na rozlíšenie homonýmnych slov treba na textových dátach previesť syntaktickú a sémantickú analýzu. [2].
- Zámená - V texte sa veľmi často nachádzajú slová obsahujúce odkazy na objekty z nasledujúceho alebo predchádzajúceho významu. Tieto odkazy sú zväča (hoci nie nutne), reprezentované pomocou zámen. Napríklad vo vete „Nadobudnuté vedomosti vývojárov by aj po ich výpovedi firme zostali.“ zámeno „ich“ v tomto prípade odkazuje na vývojárov. Vektor termov v takejto vete by namiesto lemy zámena ich, teda tvaru on, mal obsahovať dvakrát slovo vývojár. Dodať zámenu konkrétnu frázu alebo význam, na ktorý sa toto zámeno odkazuje, však vyžaduje komplexnú syntaktickú a sémantickú analýzu textu.[2]
- Významu textu - Ak by konrétny analyzovaný dokument obsahoval vetu „Tento text sa netýka športu“, tak by reprezentácia obsahu tohto textu nemala zahŕňať termy šport. Takýto level porozupenia významu textu sa prakticky nedá úplne dosiahnut a vyžaduje si úplnú sémantickú a pragmatickú analýzu, s využitím znalostí mimojazykových skutočností.[2]

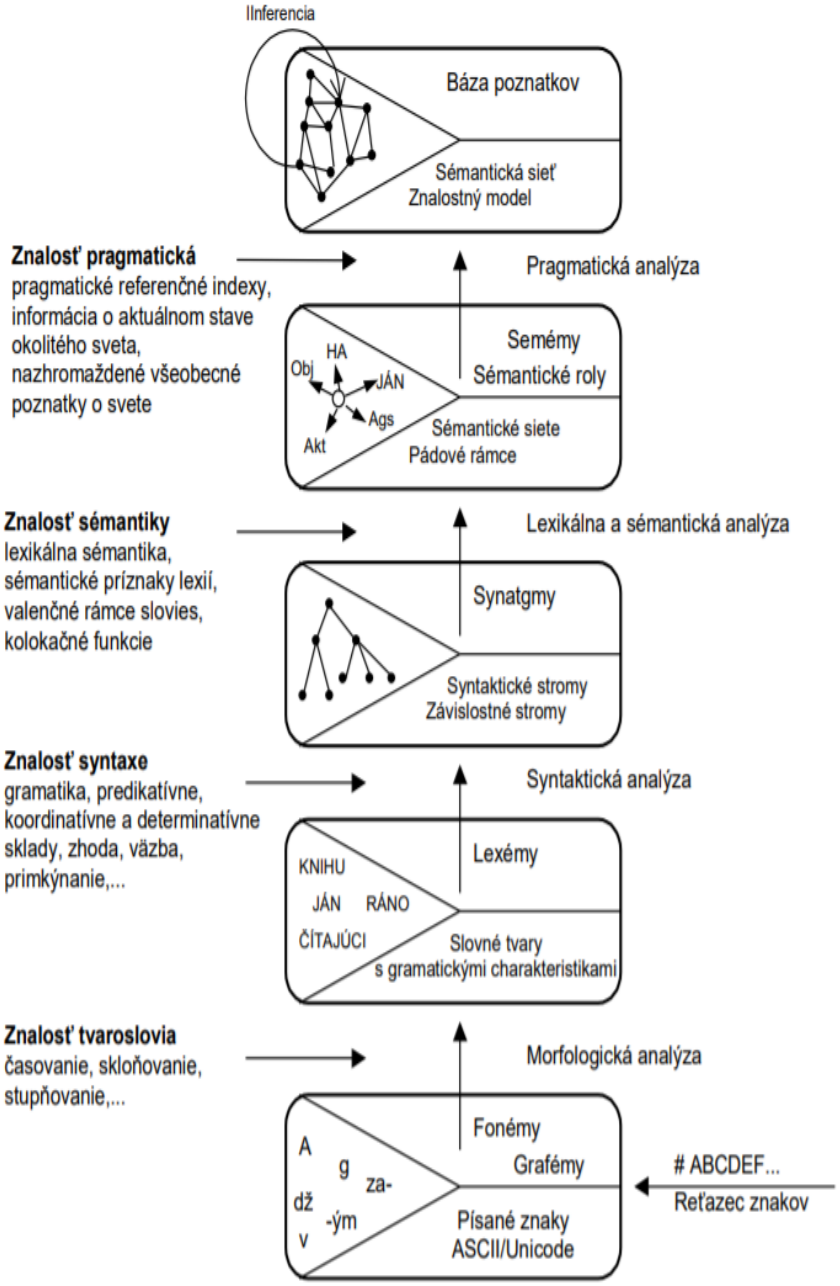
- Frázy - Slovné spojenia so špecifickým významom, napríklad Vysoká škola báňská - Technická univerzita Ostrava, kontextová analýza, výchova športu, atď. Majú z hľadiska významu samostaných slov svoj vlastný charakter. Ich celkový význam je ale iný ako ten z ktorých sa tieto výrazy skladajú. Na rozlíšenie fráz sa používajú viacslovné termy vďaka ktorým je možne viacslovné ustálené pomenovania reprezentovať. Identifikácia fráz v texte, vrátane prislúchajúcich gramatických modifikácií, predpokladá integrovanú morfologickú a syntaktickú analýzu.[2]

Štatistický prístup nehľadá podrobné vzťahy medzi jednotlivými členmi v texte ako v lingvistickom prístupe. Benefit je ale rýchlosť akou dokáže spracovať veľké objemy dát(dokumentov, textov).

2.4.2 Lingvistický prístup k analýze textu

V [2] sa lingvistický prístup opisuje ako hĺbková analýza textovej štruktúry kde sa využíva lingvistický opis ktorý dotvára model prirodzeného jazyka do veľkých detailov. V štatistickom prístupe sa vychádza z predpokladu, že slová v texte reprezentujú jeho celkový význam čize kontext textu reprezentujú plnovýznamové slová obsiahnuté v takomto texte. Takáto premisa nie je úplne správna, pretože obsah textu môže mať úplne odlišný význam od samotných slov ktoré v texte sú. Pri predspracovaní sa teda musí zohľadniť aj kontext a jeho vzťahy. V opačnom prípade sa to ukáže negatívne na úspešnosti algoritmov ktoré hľadajú informácie v textoch. V prirodzenom jazyku sa dajú hľadať a zohľadniť javy, ktoré sú dôležité z hľadiska kontextu a reprezentujú obsah textu.

Proces lingvistickej analýzy dokumentu sa zvyčajne prevádza v sekvencii. Postupne analýza začína od najnižšej, fonologickej roviny, až po poslednú sémantickú a pragmatickú rovinu. Výsledky na každej rovine sú zároveň vstupom na o úroveň vyššej. Takáto postupná jazyková analýza sa nazýva sekvenčná stratifikačná jazyková analýza. Jej model pre hľadanie kontextu v textoch je zobrazený na obrázku nižšie.



Obr. 1: Sekvenčný stratifikačný model jazykovej analýzy[2]

2.4.3 Kontextové spracovanie textu

Kontext identifikuje súvislosť medzi členmi textu. Takéto členy sa nenachádzajú v rovnakej vete ale na seba nadväzujú. Kontextové spracovanie textu priradzuje jednotlivé frázvy z jazyka k reálnym objektom. Vyžaduje, aby stroj obsahoval navyše paralelný model vonkajšieho sveta. Tento prístup vznikal v ronakom čase ako štatistický prístup ale vždy bol v jeho tieni a v praxi sa moc nevyužíval [3].

2.5 Metódy pre analýzu textu

2.5.1 Lematizácia a morfologická analýza

Bežne sa v texte jednotlivé slová vyskytujú v rôznych morfologických tvaroch, je nutné ich prevádzať na základné tvary, tzv. lemy čize na základný "slovníkový" tvar tokenu. Pri substantívach a adjektívach je to prvý pád jednotného čísla, pri slovesách neurčitok. Tento proces, ktorý z tvaru slova v texte určí základný tvar sa nazýva lematizácia.[2] Pravidlové a slovníkové algoritmy na izoláciu koreňa sú silne závislé na použitom jazyku. V angličtine, kde je izolácia koreňa pomerne jednoduchá, je najpoužívanejším Porterov algoritmus. Je založený na odstraňovaní prípon, pričom využíva pevný zoznam prípon a niektoré ďalšie pravidlá morfológie anglického jazyka. V slovenčine, češtine a ďalších jazykoch je situácia komplikovanejšia, pravidlá morfológie sú podstatne zložitejšie a izolácia koreňa sa rieši morfologickou analýzou a komplexným lingvistickým prístupom [1]. Výsledky bývajú presnejšie, avšak algoritmus je zložitejší a výrazne výpočtovo náročnejší.

2.5.2 Eliminácia neplnýznamových slov

Všetky identifikované a lematizované tokeny sú kandidátmi pre termy vektorového modelu(text). Termy(klúčové slová) vyjadrujú obsah daného dokumentu. Zároveň, je pomocou klasifikačných a zhlukovacích algoritmov výhodné minimalizovať počet termov popisujúcich jednotlivé dokumenty. Preto je nutné eliminovať termy s nízkym príspevkom k obsahu. Hlavnými nositeľmi obsahu textu sú plnovýznamové slová.Tieto lematizované tokeny sú vhodný kandidáti na reprezentáciu obsahu textu - termy. Pri neplnovýznamových slovách sa predpokladá nulový prínos k obsahu textu(spojky, predložky, zámená, častice). Sú to "stop-slová", ktoré sa v texte vyskytujú s väčšou frekvenciou, avšak nedávajú žiadnu informáciu o výslednom obsahu textu. Tokeny, ktoré vznikli z týchto neplnovýznamových slov je nutné vylúčiť zo zoznamu termov [1], [2].

Zvyčajne sa stop-slová odstraňujú dvoma základnými metódami. Prvá metóda používa tzv. negatívny slovník je to vopred zostavený zoznam neplnovýznamových slov. Z tokenizovaného textu sa odstránia slová nachádzajúce sa v negatívnom slovníku. Efektivita tohto prístupu závisí od úplnosti slovníka a jeho nevýhodou je to, že je závislý na použitom jazyku.

Druhá, automatická metóda. Eliminujú sa iba tie tokeny, ktoré sa v spracovávanom texte nachádzajú s príliš veľkou a aj príliš malou frekvenciou. Podľa [2] majú veľkú váhu a prínos k obsahu tie slová, ktorých frekvencia výskytu v celom korpuse patrí do intervalu $< \frac{N}{100}, \frac{N}{10} >$ kde N je počet dokumentov v celom korpuse. Tento spôsob je použiteľný pre väčšinu jazykov, vykazuje ale labšie výsledky ako prvý prístup.V praxi sa optimálne výsledky dosahujú kombináciou obidvoch prístupov [1].

2.5.3 Váhovanie a normovanie termov

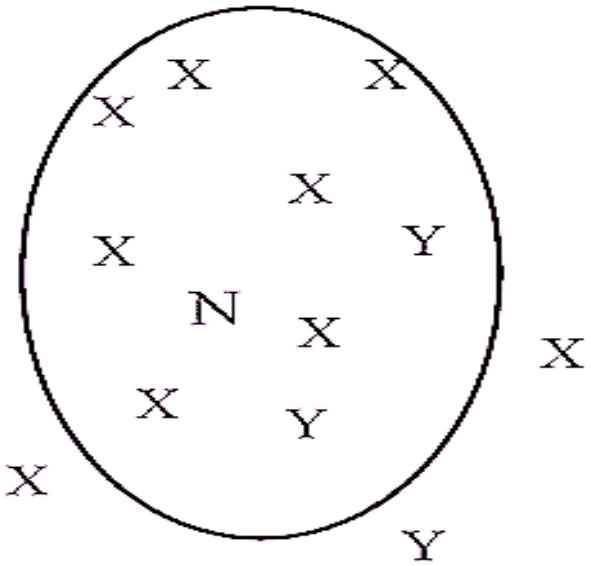
Identifikované a lematizované termy v dokumente je potrebné ohodnotiť charakteristikami vyjadrujúcimi dôležitosť termu v rámci daného dokumentu aj v rámci korpusu ako celku. Váhovanie a normovanie termov zabezpečuje dodatočné ohodnotenie a dovoľuje tým zvýšiť efektívnosť procesu kontextovej analýzy, t.j. zhlukovanie, klasifikáciu, vyhľadávanie informácií. Váhovenie je úprava frekvencie termov každého dokumentu ktorý sa nachádza v korpuse dokumentov. Toto váhovanie prebieha v dvoch rovinách. Lokálne, váhovanie na základe počtu výskytov v samotnom dokumente. Globálne, váhovanie na základe frekvencie výskytov t-tého termu v celom korpuse. Váhovaná frekvencia výskytu t-tého termu je daná súčinom lokálnej a globálnej váhy.

2.6 Metódy pre klasifikáciu textov

2.6.1 K–najbližších susedov

Pri učení klasifikátorov založených na pravidle k–najbližších susedov sa len odpamätajú všetky trénovacie príklady z D (tieto algoritmy označujú ako „lazy“). Pri klasifikácii nového dokumentu sa podľa funkcie podobnosti určí k najpodobnejších dokumentov z D a klasifikátor zaradí nový dokument do triedy určenej podľa klasifikácie susedov.
Najdôležitejšou časťou algoritmu je metrika na meranie vzdialenosti vo vektorovom priestore dokumentov(funkcia podobnosti). Keďže dokumenty sú reprezentované ako reálne vektory, je možné použiť Euklidovskú metriku, ale významovej podobnosti dokumentov lepšie zodpovedá kosínusová funkcia podobnosti. Je možné použiť aj podobné funkcie navrhnuté pre vyhľadávanie, kde funkcia podobnosti určuje skóre dokumentu voči dopytu.
V prípade, že má byť príklad klasifikovaný iba do jednej triedy, vyberie sa najčastejšie sa vyskytujúca trieda. Okrem toho môže byť „hlasovanie“ jednotlivých susedov vážené podľa ich vzdialenosti voči klasifikovanému dokumentu, t.j. trieda najbližšieho suseda má najväčšiu váhu atď. Ak je potrebné klasifikovať dokumenty do viacerých tried, určí sa celkové skóre (suma pre všetkých susedov) pre každú triedu c_j a ak je skóre väčšie ako zvolená prahová hodnota, príklad sa zaradí do c_j. Hodnota parametra k sa určuje testovaním na validačnej množine príkladov. Pri krížovej validácii sa trénovacia množina D rozdelí na n podmnožín D_1, ..., D_n, najprv sa naučí klasifikátor na dátach z množín D_1 ...,D_{n-1} a otestuje sa. Postupne sa otestujú všetky

podmnožiny.
Najväčšou nevýhodou tohto algoritmu je veľká pamäťová a časová náročnosť pri klasifikácii, keďže je potrebné uchovať a vypočítať vzdialenosť voči všetkým trénovacím príkladom[1].



Obr. 2: Príklad použitia metódy K–najbližších susedov

Obrázok zobrazuje okolie nového prípadu označeného písmenom N. V okolí sa nachádzajú prípady zo skupín X a Y. Keďže počet prípadov zo skupiny X je väčší ako zo skupiny Y, tak prípad N bude zaradený do skupiny X.

2.6.2 Naivný Bayesov Klasifikátor

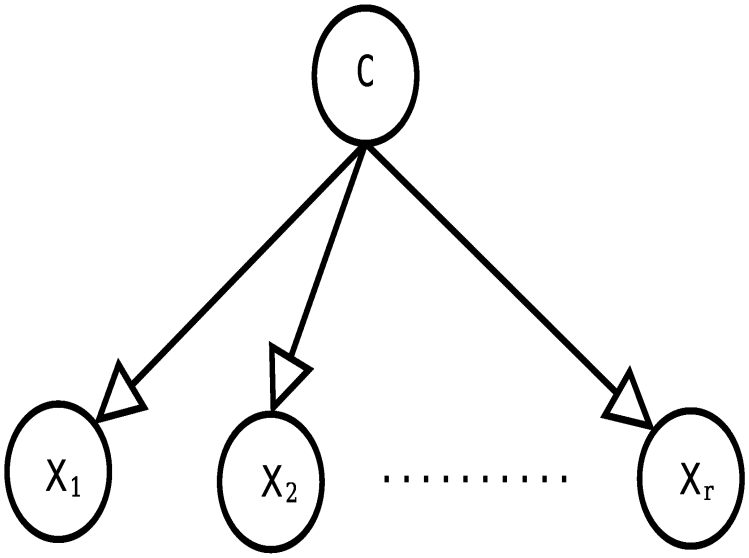
Naivný Bayesov klasifikátor klasifikuje nové dokumenty na základe podmienenej pravdepodobnosti *P(*c_j*/ d)*, tj. pravdepodobnosti, že dokument d má byť zaradený do kategórie c_j. Pri učení klasifikátora je na vstupe trénovacia množina dokumentov $D = d_1, ..., d_{|D|}$, ktoré sú klasifikované do množiny tried $C = c_1, \ldots, c_{|C|}$. Každý z dokumentov *di* je reprezentovaný ako postupnosť termov, ktoré sa vyskytli v dokumente $di = <t_i, 1, \ldots, t_i, |d| >$, kde /*d*/ je dĺžka dokumentu d_i Na základe trénovacej množiny D nie je možné určiť odhad podmienenej pravdepodobnosti *P(*c_j*/ d)* pre neznámy dokument d priamo. Je však možné priamo určiť pravdepodobnosť kategórie P(c_j) a pravdepodobnosť výskytu termu t v dokumente za predpokladu že dokument patrí do kategórie c_j, t.j. *P(t / *c_j*)*. Naivný Bayesov klasifikátor využíva odhad týchto pravdepodobností určených podľa trénovacej množiny dokumentov pre určenie pravdepodobnosti *P(*c_j*/ d)*. V prvom kroku je potrebné skombinovať pravdepodobnosti *P(t / *c_j*))* jednotlivých termov vyskytujúcich sa v dokumente tak, aby sa získal odhad pravdepodobnosti *P(d / *c_j*)* pre celý dokument. Za predpokladu, že každý term sa vyskytuje v dokumente štatisticky nezávisle od ostatných termov je možné určiť *P(d / *c_j*)* ako súčin:

$$P(d|c_j) = \prod_{i=1}^{|d|} P(t_i|c_j)$$

Pravdepodobnosť *P(*c_j*/ d)* je potom určená z *P(d / *c_j*)* podľa Bayesovho pravidla:

$$P(c_j|d) = \frac{P(c_j)P(d|c_j)}{P(d)}$$

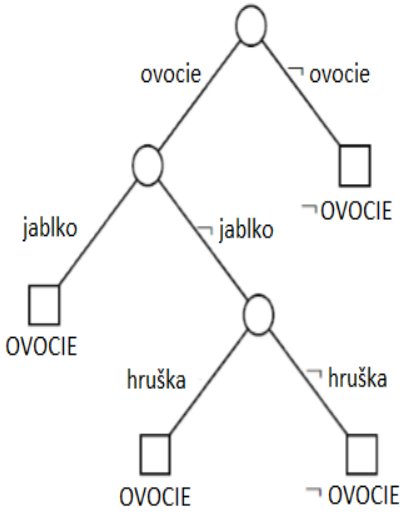
Ak má byť dokument klasifikovaný iba do jednej triedy, na základe Bayesovho pravidla sa vyberie trieda, pre ktorú je pravdepodobnosť *P(*c_j*/ d)* maximálna. Pravdepodobnosť *P(d)* je možné v tomto prípade vynechať, keďže neovplyvňuje rozhodovanie podľa maximálnej hodnoty *P(*c_j*/ d)*. Pre viacnásobnú klasifikáciu je dokument zaradený do triedy c_j, ak pravdepodobnosť*P(*c_j*/ d)* prekročí zvolenú prahovú hodnotu, napr. 0,5.[2]



Obr. 3: Naivný Bayesov Klasifikátor

2.6.3 Rozhodovacie stromy

Na obrázku nižsie je príklad rozhodovacieho stromu. Každý nelistový uzol stromu je označený testom, ktorý rozdeľuje dokumenty podľa výskytu jedného termu. Listové uzly sú označené priradením triedy. Klasifikácia prebieha rekurzívne od koreňového uzla, zvolením vetvy podľa testu až pokiaľ sa nedosiahne listový uzol, t.j. pre uvedený príklad, dokumenty sú zaradené do triedy OVOCIE ak obsahujú term „ovocie" a zároveň obsahujú term „jablko" alebo „hruška". Rozhodovacie stromy sú učené metódou "zhora nadol". Počiatočný strom je tvorený iba jedným uzlom, ktorý pokrýva všetky trénovacie dokumenty. Ak nastane prípad, že všetky príklady pokryté daným uzlom majú rovnakú triedu, delenie množiny dokumentov nie je potrebné a uzol sa označí ako listový. Inak algoritmus priradí uzlu logický test, ktorý rozdelí príklady na disjunktné podmnožiny, pre ktoré sa vytvoria nové uzly stromu. Celý proces sa potom rekurzívne opakuje na nových potomkoch, až pokiaľ sa nedosiahne úplné oddelenie príkladov jednotlivých tried.[3]

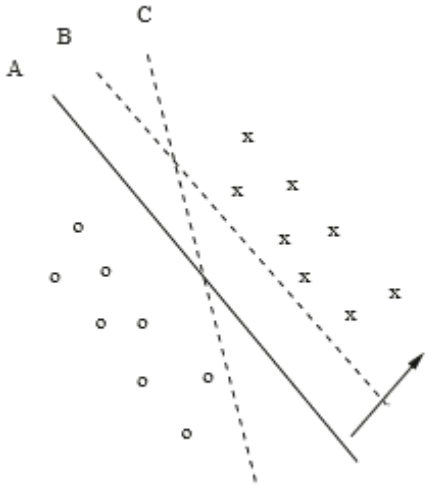


Obr. 4: Rozhodovací strom

Výber testu pre nelistové uzly je založený na hodnotiacej funkcii, ktorá zodpovedá "variabilite"tried v podmnožinách vytvorených po rozdelení uzla. Pri delení sa otestujú všetky možné testy a vyberie sa test, ktorý vedie k čo najväčšej redukcii hodnotiacej funkcie a teda k čo najväčšiemu počtu príkladov z jednej triedy v každej vytvorenej podmnožine.

2.6.4 Support Vector Machines

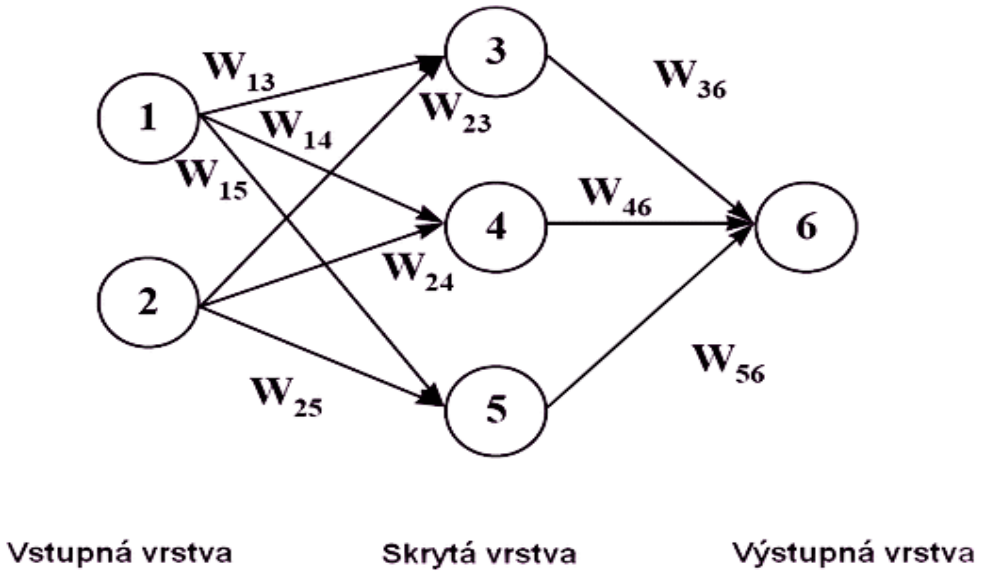
Support Vector Machines je metóda založená na strojovom učení. Princípom SVM je rozdelenie tréningových dát zakreslených v bodovom diagrame na dve protiľahlé oblasti patriace jednotlivým triedam dát. SVM priestor rozdeľuje nadrovinou. Nadrovina tiež môže byť označená ako tzv. rozhodovacia hranica, ktorá oddeľuje na bodovom diagrame dve triedy a určuje, ktoré body patria do ktorej triedy.[1] Pre optimálnu nadrovinu platí, že musí byť umiestnená v čo najväčšom odstupe od krajných bodov, nazývaných podporné vektory. Inými slovami, nadrovina musí vytvárať okolo seba čo najväčšie pásmo bez bodov a musí byť v strede tohto pásma tak, aby okrajové body oboch rozdelených oblastí boli rovnako vzdialené. SVM sa delí na lineárnu a nelineárnu, a to podľa schopnosti rozdeliť priestor.



Obr. 5: Rozhodovacie hranice, optimálna rovina A

2.6.5 Neurónové siete

Neurónová sieť je jedným z výpočtových modelov používaných v umelej inteligencii. Jej chovanie sa nápadne podobá na ekvivaletné biologické štruktúry. Umelá neurónová sieť je štruktúra určená pre distribuované paralelné spracovanie údajov. Skládá sa z umelých neurónov, ktorých predobrazom je biologický neurón. Neuróny sú navzájom prepojené, navzájom si odovzdávajú signály a transformujú ich pomocou určitých prenosových funkcií. Neurony majú ľubovoľný počet vstupov, ale iba jeden výstup.[14] Neurónová sieť začína vstupnou vrstvou, v ktorej každý uzol predstavuje vstupnú premennú. Uzly z tejto vrstvy sú spojené s uzlami zo skrytej vrstvy. Každý uzol zo vstupnej vrstvy je spojený s každým uzlom zo skrytej vrstvy. Uzly zo strednej vrstvy môžu byť spojené s uzlami z ďalšej strednej vrstvy alebo môžu byť spojené s výstupnou vrstvou. Výstupná vrstva sa skladá z jednej alebo viacerých výstupných premenných.



Obr. 6: Príklad neurónovej siete

Okrem uzlov na vstupnej vrstve, každý uzol má niekoľko vstupov. Tieto vstupy sú vynásobené váhou prepojenia Wxy a následne sčítané. Na výslednú hodnotu sa aplikuje aktivačná funkcia pričom výsledná hodnota sa považuje za výstupnú hodnotu uzla. Táto sa pošle na vstup uzlom na ďalšej vrstve. Aktivačné funkcie sú definované pri tvorbe takejto siete. Problémom je stanovenie váh jednotlivých prepojení medzi uzlami. Sú stanovené tréningovými metódami, ktorých je niekoľko a je ich možné nájsť v príslušnej literatúre k neurónovým sieťam.

Neurónové siete môžu byť využívané pre kategorizáciu dokumentov. V takom prípade sú váhy rysov vstupy pre počiatočné uzly. Vzťahy a závislosti medzi neurónmi vnútri siete potom určujú závislosti medzi slovami a kategóriami. Pre kategorizáciu dokumentu sú váhy rysov načítané na vstupné uzly. Potom sa jednotlivé uzly aktivujú a vyhodnocujú. Výsledky sa potom propagujú celou sieťou a výstupný uzol celej siete determinuje príslušnosť dokumentu do určitej kategórie.

3 Zdroj dát

Pre potreby trénovania a testovania kategórií je potrebná databáza dokumentov, korpus, ktorá bude obsahovať veľké množstvo textu s už známou kategóriou. V tejto práci používam dva takéto zdroje. Pre slovneský jazyk je to Slovenský národný korpus a pre anglický jazyk Wikipedia.

3.1 Korpus

K tomu, aby bol algoritmus schopný pracovat s kontextom, musí mať k dispozici vhodný dátový zdroj, z kterého bude čerpat. Takémuto zdroju sa hovorí korpus.

V jazykovej lingvistike sa pod pojmom korpus rozumie systematicky zozbieraný súbor písaných alebo hovorených prejavou daného jazyka (prípadne viacjazyčných tzv. paralelných korpusov aj viac jazykov), ktorý splňuje stanovené podmienky. Podmienkami sú podľa[6] reprezantatívnosť, vzorkovateľnosť, strojovo čítateľná forma, jasne vymedzený rozsah, referencie podľa štandartov. Je ich možné deliť podľa osbahu a rozsahu na kvantitatívne a kvalitatívne, podľa formy na písané a hovorené korpusy, z hľadiska času na synchrónne a diachrónne. Tie najvetšie korpusy majú rozsah v stovkách až tisíckach miliónov slov (SNK - 1 160 286 731 slov, verzia prim-8.0-public-all), bývajú spravidla morfologicky, syntakticky, sémanticky, druhovo anotované. Na ich prehľiadanie existujú aplikácie tzv. korpusové manažéry. Hlavné využitie takýchto korpusov je v lexikologii a lexikografii, nad dátmi sú vytvárné slovníky, prekladače. Z neterminologického hľadiska je možné za korpus na základe etymológie považovať každý textový súbor uskladnený alebo používaný ku konkrétnemu účelu, v lingvistike je to zvyčajne k empirickému jazykovědnému bádaniu. V mojej práci teda korpus chápem práve ako súbor textov v prirodzenom jazyku, ktorý bude popísaným metódam slúžiť ako zdroj dát a nad ktorým budem vykonávať experimenty.

3.2 Slovenský Národný Korpus

Slovenský národný korpus (SNK) je elektronická databáza primárne obsahujúca slovenské texty od r. 1955 z rôznych štýlov, žánrov, vecných oblastí, regiónov a pod. v rozsahu poskytnutom autormi a majiteľmi autorských a/alebo distribučných práv na základe licenčnej zmluvy. Texty a slová v korpuse sú obohatené o jazykové informácie a predstavujú referenčný materiálový zdroj poznatkov o slovenčine a jej reálnom používaní, ktoré sa z korpusu získavajú pomocou špecializovaných vyhľadávacích nástrojov. Korpus nie je elektronická knižnica, ani nenahrádza kodifikačné príručky. V korpusových dátach sa dajú vyhľadávať jazykové informácie na výskumné, učebné a iné výlučne nekomerčné ciele [17]

Pre potreby trénovania a testovania kategórií v tejto práci som pre slovenský jazyk použil nasledovné vygenerované kolekcie.

Názov	Veľkosť	Počet znakov	Počet slov
SNK_KAT_1	410 MB	321 983 364	51 077 530
SNK_KAT_2	280 MB	248 799 402	34 861 152
SNK_KAT_3	370 MB	302 018 236	42 520 136
SNK_TEST	480 MB	375 654 752	60 104 152

Tabuľka 1: Informácie o vygenerovaných kolekciach SNK

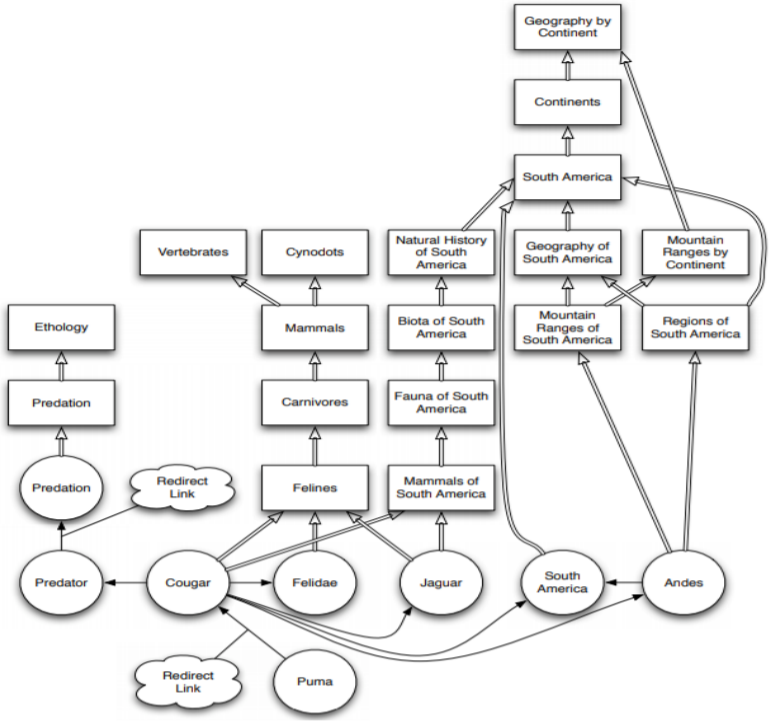
3.3 Wikipédia

Wikipedia je bezplatnou encyklopédiou a polyglotom nadácie Wikimedia Foundation (nezisková organizácia) Organizácia). Viac ako 20 miliónov článkov v 282 jazykoch a dialektoch bolo vypracovaných spoločne dobrovoľníkmi z celého sveta a prakticky každý, kto má prístup k internetu, môže byť editor. Projekt začal v januári 2001 Jimmy Wales a Larry Sanger a je najväčší a najpopulárnejší on-line zdroj informácií. Patrí medzi 10 najpopulárnejších webových stránok na svete a priniesla úspech aj pre sesterské projekty.

Prakticky všetci návštevníci môžu obsah Wikipedie upravovať a vytvárať nové články. Zmeny sú viditeľné ihneď po stlačení tlačidla Uložiť a schválené editorom. Táto podmienka bola zavedená v novembri 2008 na ochranu čitateľov pred účinkami vandalizmu. Wikipedia je postavená na presvedčení, že spolupráca medzi užívateľmi povedie k neustálemu zlepšovaniu obsahu spôsobom, akým to má bolo dosiahnuté v mnohých projektoch s otvoreným zdrojom. Niektorí redaktori článkov Wikipedia popísali editačný proces ako evolučný proces sociálneho darwinizmu.

Každý článok na Wikipédii popisuje danú tému a má krátky titul, ktorý je vystihuje podstatu článku. Každý článok patrí do aspoň jednej kategórie a hypertextové odkazy medzi články zachytávajú ich sémantické vzťahy. Konkrétne, sémantické vzťahy sú: rovnocennosť, hierarchickosť a asociatívnosť. Wikipedia obsahuje vždy práve jeden článok pre akýkoľvek koncept (tzv.

preferovný výraz). Obrázku nižšie ukazuje príklad presmerovania medzi synonymami "Pumaä "cougar". Okrem synomymov zvládajú presmerovania aj odchýlky pravopisu, skratky, hovorové pomenovania, a vedecké pojmy.[15]



Obr. 7: Príklad taxonómie Wikipédie

Každému článku na Wikipedii patrí prinajmenšom jedna kategória a kategórie sú vnorené do hierarchickej organizácie. Anglická verzia je najväčšia s približne 35% všetkých slov. Pre potreby trénovania a testovania kategórií v tejto práci som pre anglický jazyk použil nasledovné vygenerované kolekcie.

Názov	Veľkosť	Počet znakov	Počet slov
WIKI_KAT_1	215 MB	189 087 544	26 494 475
WIKI_KAT_2	230 MB	204 015 509	28 586 144
WIKI_KAT_3	207 MB	172 755 108	22 177 001
WIKI_TEST	234 MB	206 722 034	30 045 622

Tabuľka 2: Informácie o vygenerovaných kolekciach Wikipedia

4 N-gram

N-gram je definovaný ako sled po sebe nasledujúcih položiek z danej sekvencie.[4] Zo sémantického pohľadu môže byť táto postupnosť buď postupnosťou hlások, slabík, písmen alebo slov. V praxi sa častejšie vyskytujú n-gramy ako sled slov. Sled dvoch po sebe nasledujúcich položiek býva často označovaný ako bigram, pre sled troch položiek je zaužívaný pojem trigram. Od štyroch a vyššie sa používa označenie N-gram, kde N je nahradené počtom za sebou nasledujúcich elementov.

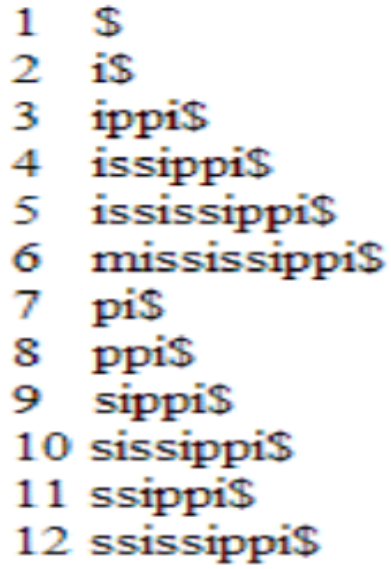
Spôsobov ako n-gramy extrahovat z koprusu je popisaných viacero. Popri modifikovaného Apriori algoritmu [4], prezentovali Kit a Wilks (1998) a následně Yamamoto a Church (2001) postup extrakice n-gramov založený na datové štruktúre sufixového poľa. Ďalsia z možností je potom použitie sufixového stromu, ktorý je popísaný v článku[5].

4.1 Algoritmy pre extrakciu N-gramov

Ako jeden z prvých publikovaných algoritmov pre extrakciu N-gramov bol algoritmus založený sufixových poliach, který se v súčstnosti používá v rôzne modifikovaný. Jednou z možností je aj použitie sufixového stromu. Podľa experimentov v [7] lepšiu výkonnosť majú algoritmy založené na suffixových poliach než implementácie nad sufixovým stromom. Ďalej sa pre extrakciu požívajú algoritmy invertovaného indexu alebo algoritmus Apriori.

4.2 Sufixové pole

Sufixové pole je v porovnání so sufixovým stromom pamäťovo efektívnjejšia dátová štruktúra, umožnuje rýchle vyhľadávanie podreťazcov v texte. Základom suffixového poľa je zoznam ukazovateľov, z ktorých každý ukazuje na pozíciu v korpuse (token), može ním byť znak alebo slovo, a ktorý virtuálne označuje postupnosť tokenov od tejto pozície až po koniec korpusu[7]. Takejto postupnosti sa hovorí podreťazec. Jedná o seřazené pole všetkých sufixov v texte. Sufixové pole pre text dĺžky n môže byť zostavené v čase *O(n2 log n), O(n log n)* alebo aj s lineárnovou časovou zložitosťou *O(n)*



Obr. 8: Sufixové pole reťazca mississippi$ ($ značí koniec reťazca)

Samotné sufixové pole nemá rovnakú vyjadrovaciu schopnost ako sufixový strom. Ak k sufixovému poli pridáme pole, ktoré neise informáciu o lcp (longest common prefix), môžeme na týchto dvoch poliach simulovat každý priechod zdola-nahor sufixovým stromom. Avšak informácie o lcp umožnujú simulovať priechod sufixovým stromom iba smerom od potomkov k rodičovským uzlom. Rozšírením sufixového poľa o ďalšie pomocné pole vďaka ktorému je možne simulovať priechody sufixovým stromom od rodičovských ulozlov k potomkom. Na základe takéhoto rozšíreného sufixového poľa bol zostavený koncept lcp-intervalových stromov. Tieto stromy nie je nutné v praxi konštruovat a pritom sú ekvivaletné k sufixovým stromom. V[10] bolo dokázané že akýkoľvek algoritmus pracujúici so sufixovým stromom môže byť prevedený na ekvivaletný algoritmus využivajúci rozšírené sufixové pole pri zachovaní časovej náročnosti.

4.3 Rozšírené sufixové pole

Prvý krát bolo predstavené v [3]. Obecne sa jedná o sufixové pole doplnené o pomocné polia. Tieto polia každopádne vyžadujú ďalšiu pamäť.

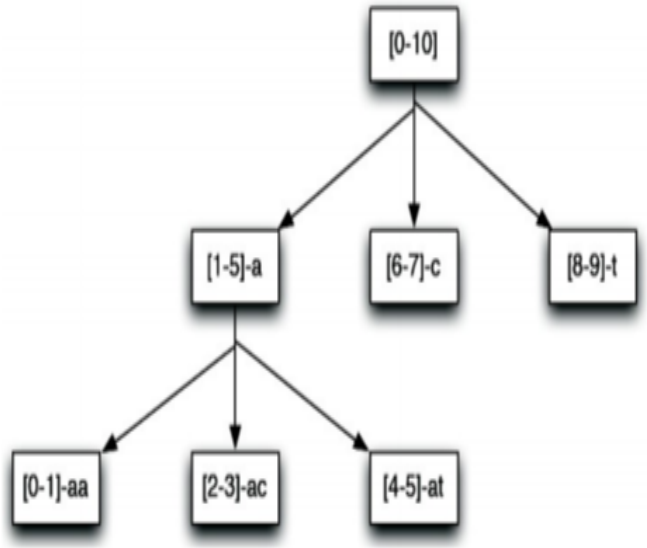
Takéto rozširujúce pole je aj *lcp* pole. Ide o pole ktoré udáva koľko spločných znakov majú susediace sufixy. Hodnoty *lcp* sú využité k definovaniu intervalov - lcp intervaly. Takýto interval sa dá predstaviť ako interval ktorý korešponduje určitému rozsahu sufixov so špecifickým prefixom. Interval $[i...j], 0 \leq i < j \leq n$, kde n je dĺžka sufixového poľa, je lcp intervalom pre lcp hodnoty l ak sú splnené podmienky (lcparr označuje lcp pole):

1. lcparr[i] < l,
2. lcparr[k] ≥ l pro všetky k, pre ktoré platí i+1 ≤ k ≤ j,

3. lcparr[k] = k aspoň pre jedno k, pre ktoré platí i+1 ≤ k ≤ j,

4. lcparr[j+1] < l.

Lcp intervaly v môžu v sebe obsahovať menšie lcp intervaly. Na tieto vnorené intervaly je možné pozerať ako na stromovú štruktúru - lcp intervalový strom. Takáto štruktúra je implicitná a naviac má rovnákú štruktúru ako sufixový strom.



Obr. 9: Lcp intervalový strom nad reťazcom S = acaaaacatat$

4.4 Sufixový strom

Sufixový strom je dátová štruktúra, ktorá umožňuje veľmi efektívnu prácu s reťazcami. Je vhodná pre prácu aj s veľkými objemami dát. Používa sa pre vyhľadávanie reťazcov, hľadanie najčastejšie opakovaných podreťazcov, pre nachádzanie palindromov či pre vyhľadávanie najdlhších podreťazcov. Koncept sufixového stromu pochádza zo 70. rokov, kedy ho predstavil pán Weiner, ale až v polovici 90. rokov bol modifikovaný pre vyhľadávania fráz. Publikované články uvádzajú zložitosť algoritmu *O (n)*. Jedná sa teda o lineárnu zložitosť závislú predovšetkým od veľkosti vstupnej kolekcie dokumentov. Výhodou Sufixového stromu je tiež nezávislosť na poradie spracovávaných dokumentov a jazyková nezávislosť umožňujúci spracovanie kolekcií obsahujúcich dokumenty v rôznych jazykoch.[7, 9]

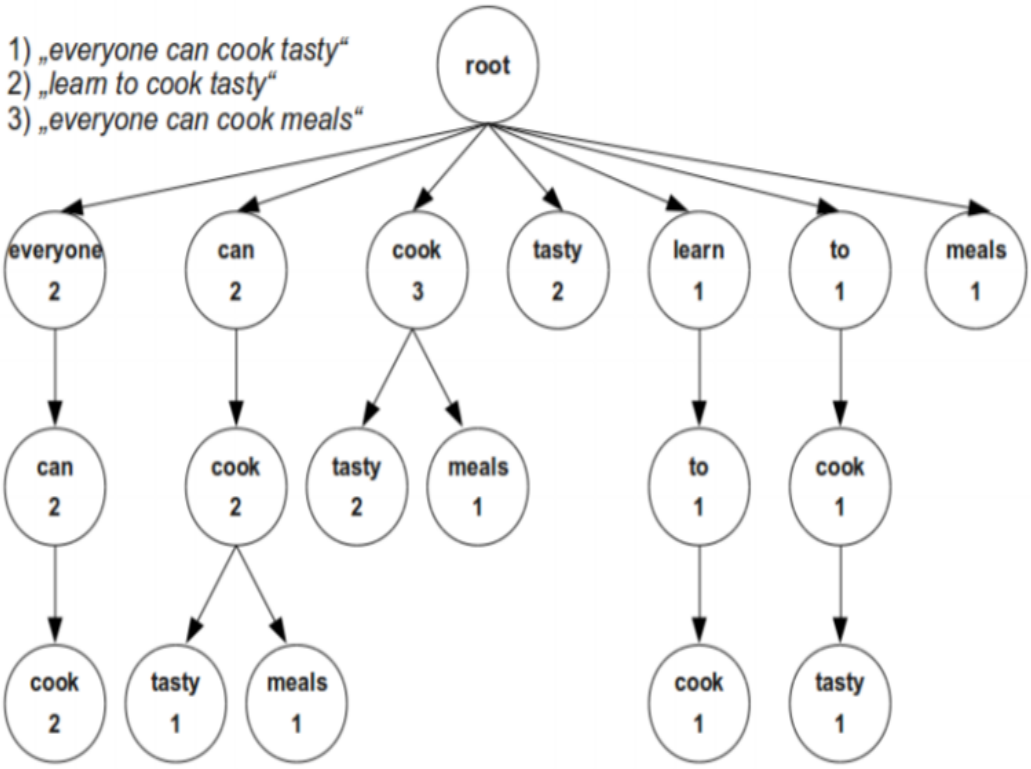
Obecne sa dá princpíp extrakcie N-gramov predstavit ako okno o velkosti slov *N*, ktoré sa posúva po slovách z textu. Výsledná hodnota • teda predstavuje maximálnu dĺžku hľadaných N-gramov. Rovnako je ním stanovená aj maximálna hĺbka Suffixovéh ostromu. Algoritmus pre získanie N-gramov prebieha nasledovne:

1. Vytvorí sa koreňový uzol Sufixového stromu - žiadne slovo

2. Prečíta sa N prvých slov s_1 ... s_N zo vstupného dokumentu (ak nie je dostatok slov - musí sa čítať vždy slová nasledujúce bezprostredne po sebe a nie je možné presahovať hranice jednotlivých viet či súvisiacich celkov).

3. Každé slovo s_i, kde $i = 1 \ldots N$, je vložené do Sufixového stromu a to takým spôsobom, že každé slovo s_i je vložené do i-tého stupňa zanorenia v štruktúre stromu. Ďalej cesta od koreňového uzla k práve spracovávanému slovu si musí viesť skrze uzly reprezentujúce slová s_1 ... s_{i-1}. Ak uzol reprezentujúci spracovávanej slovo si už existuje, tak sa iba inkrementuje počet výskytov daného uzla.

4. Vymaže sa prvé slovo zo vstupného dokumentu a ak už nie je vstup prázdny, tak sa pokračuje znovu od kroku 2.



Obr. 10: Sufixový strom

4.5 Algoritmus pre zostrojenie sufixového stromu

Základným prvkom štruktúry je koreň stromu (tzv. Root node), ktorý nereprezentuje žiadne slovo, ale obsahuje odkazy na všetky uzly o hĺbke jedna, ktorými sú vlastne všetky unikátne rysy v dokumente. Vďaka tomu je stromová štruktúra rozsiahla do šírky. Strom nedosahuje veľkého rozsahu do hĺbky, pretože maximálna hĺbka vetvy stromu je rovnaká ako maximálna

stanovená veľkosť N-gramu. Z implementačného hľadiska je dobré využívať hash tabuľku pre rýchle vyhľadávania v uzloch.

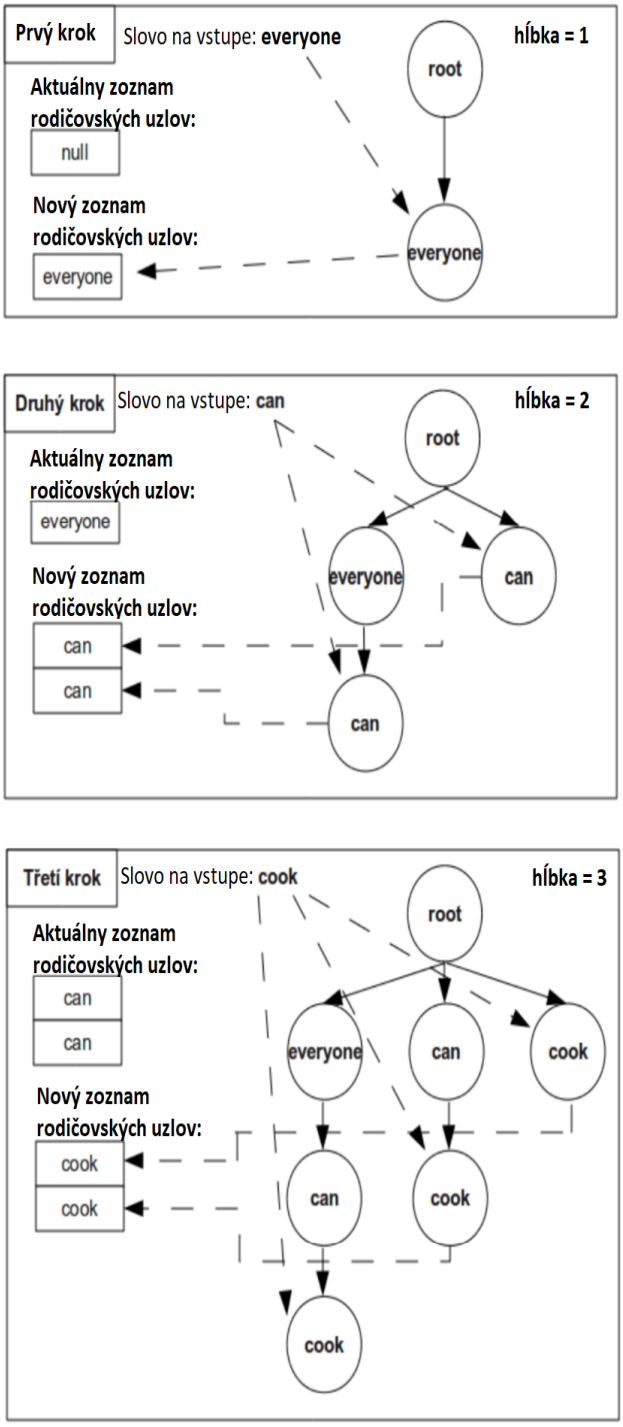
Ako bolo spomenuté, najvyššie v strome stojí root node, ktorý má v sebe hash tabuľku s odkazmi na uzly potomkov. Každý takýto uzol potom obsahuje slovo, ktoré reprezentuje počet jeho výskytov v dokumente a opäť hash tabuľku s odkazmi na svojich potomkov. V hash tabuľke sa používa ako kľúč slovo, ktoré uzol reprezentuje a položkou je ukazovateľ na objekt dcérskeho uzla.

N-gramy zo sufixového stromu extrahujeme tak, že strom prechádzame po jednotlivých vetvách. Dcérske uzly koreňového uzla stromu obsahujú unigramy, teda N-gramy o veľkosti jedna. Uzly v hĺbke dva obsahujú bigramy, ktoré sú zostavené zo slova nadradeného uzla a slová reprezentovaného daným uzlom v hĺbke dva. Takto sa zostavujú N-gramy až do maximálnej hĺbky vetvy stromu.

Najprv algoritmus rozdelí text dokumentu na jednotlivé slová, ktoré sú následne uložená do zoznamu. V texte sú zachované na konci viet bodky, ktoré sú uložené ako samostatné prvky v zozname slov. Je to preto, aby algoritmus poznal hranice viet a pri generovaní N-gramov ich nepresahoval. Tento zoznam slov slúži ako vstup pre vytvorenie štruktúry Sufixového stromu. Potom sa začnú postupne čítať slová zo vstupného súboru. Pri načítaní prvého slova je vytvorený odkaz v koreňovom uzle na synovský uzol reprezentujúci dané slovo. Tiež sa referencie na vzniknutý uzol pridá do zoznamu rodičovských uzlov.

Potom sa čítajú nasledujúce slová zo vstupného zoznamu. Pri načítaní každého slova sa vykoná kontrola, či už uzol reprezentujúci rovnaké slovo nie je v hash tabuľke koreňového uzla obsiahnutý. Pokiaľ je, tak sa iba inkrementuje počet jeho výskytov a pokiaľ nie je, vytvorí sa nový dcérsky uzol koreňového uzla reprezentujúci dané slovo. V každom prípade je referencia na inkrementovaný či vytvorený uzol pridaná do zoznamu rodičovských uzlov.

Okrem pridávania dcérskeho uzla do koreňového uzla stromu sa ešte prechádza zoznam rodičovských uzlov, ktorý bol vytvorený pri spracovávaní predchádzajúceho slova. Pre aktuálne spracovávané slovo sa skontroluje v každom zázname v zozname rodičovských uzlov, či už nemajú potomka reprezentujúceho rovnaké slovo. Ak áno, zvýši sa iba počet jeho výskytov. V opačnom prípade sa vytvorí nový dcérsky uzol a odkaz na neho je pridaný do referencií nadradeného uzla zo zoznamu rodičovských uzlov. Pri novom uzle sa skontroluje jeho hĺbka v štruktúre stromu a pokiaľ nedosiahla maximálnej nastavenej hĺbky, pridá sa uzol do zoznamu rodičovských uzlov pre ďalšie spracovávané slovo. Výnimkou je tiež situácia, kedy je načítaný miesto slova znak bodka ukončujúci vetu. V takomto prípade sa vymaže aktuálny zoznam rodičovských uzlov a spracovávajú sa ďalšie slová rovnakým spôsobom.



Obr. 11: Algoritmus pre zostrojenie sufixového stromu

4.6 Algoritmus invertovaného indexu

Všeobecne sa jedná o indexovaciu štruktúru, ktorá slúži pre mapovanie slov k ich lokáciam v dokumentoch či v množinách dokumentov. Invertovaný index umožňuje jednoduché a veľmi rýchle vyhľadávanie. Primárne sa používa pre indexáciu dokumentov na Internete. Tento algoritmus nebol navrhnutý pre extrakciu N-gramov a pre toto využitie je nutné ho čiastočne modifikovať

Algoritmus najskôr vytvorí štruktúru invertovaného indexu a vloží do nej všetky slová, vyskytujúce sa v dokumente, s referenciou na ich umiestnenie. S každým slovom je pri vkladaní do štruktúry vykonávaná rovnaká operácia [9]:

1. Najprv sa nájdu všetky výskyty spracovávaného slova a vytvoria sa zoznamy všetkých slov, ktoré za ním nasledujú na rôznych miestach v dokumente.
2. Zoznamy sa zoradia a spočítajú sa výskyty jednotlivých slov v dokumente.
3. Slová zo zoznamov, ktoré sa v dokumente nevyskytujú viac ako dvakrát, sú vymazané (Tento krok sa môže vynechať, ak požadujeme úplnú množinu N-gramov).
4. Duplicitné zoznamy sú vymazané a k unikátnym je uložený ich počet výskytov.
5. Podreťazce, ktorými zoznamy začínajú, sú pridané ako nové zoznamy a opäť sa vykoná prepočítanie výskytov a redukcia duplicít.
6. Nakoniec sú nájdené zoznamy pridané medzi získané N-gramy.

Experimenty vykonané v [9] ukazujú, že použitie Sufixového stromu je výrazne výpočtovo efektívnejšie. Podľa testov bol Invertovaný index mnohonásobne pomalší a v závislosti na veľkosti vstupnej kolekcie sa jeho výpočtový čas zhoršoval. Nezanedbateľnov výhodou invertovaného indexu sú ale pamäťové nároky, ktoré sú zhruba polovičné oproti Sufixovému stromu.

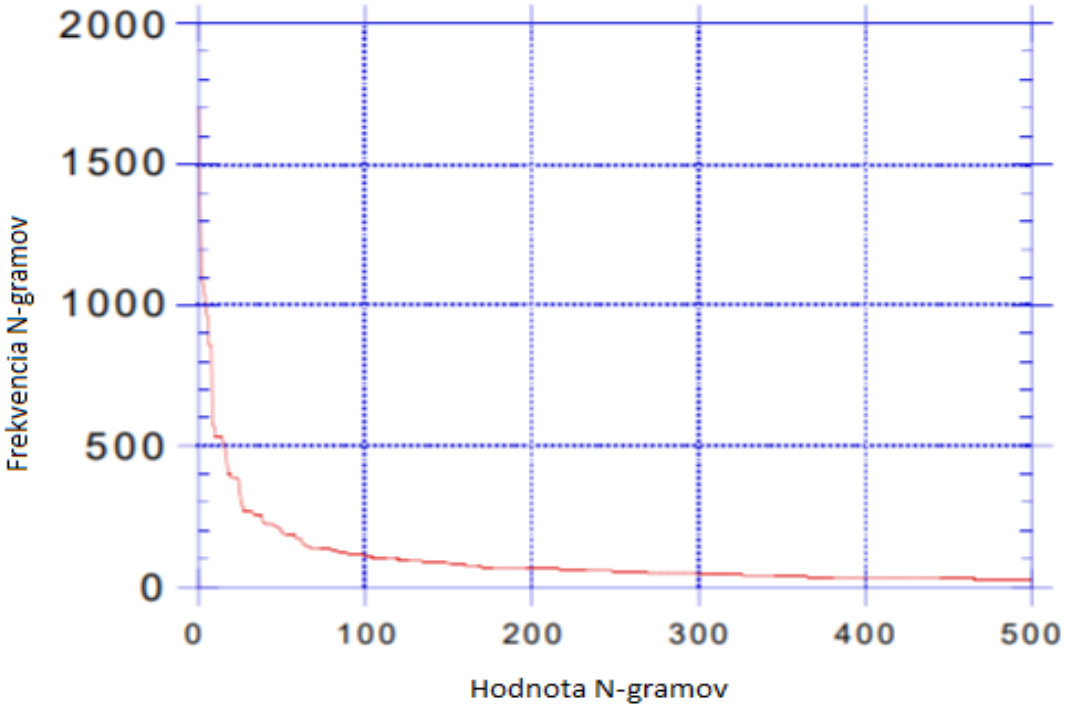
5 Kategorizácia

5.1 Kategorizácia prostredníctvom extrakcie N-gramov

Ľudské jazyky majú vždy nejaké slová ktoré sa vyskytujú častejšie než iné. Jeden z najbežnejších spôsobov ako vyjadriť túto myšlienku je zákon známy ako Zipfov zákon [11].

Zipfov zákon hovorí, že ak si vezmeme korpus nejakého prirodzeného jazyka, potom frekvencia výskytu nejakého slova je nepriamo úmerná jeho pozícii vo frekvenčnej tabuľke. Najčastejšie slovo sa vyskytuje približne dvakrát tak často ako druhé najčastejšie, trikrát tak často ako tretie, atď. Napr. v korpuse anglického jazyka (Brown Corpus) „the“ je najčastejšie a jeho výskyt je približne 7%. V súlade s Zipfovým zákonom výskyt „of“ je niečo vyše 3.5% of words, atď. Iba 135 slov tvorí asi polovicu Brownského korpusu.

Dôsledkom tohto zákona je, že existuje vždy súbor slov, ktoré dominujú nad ostatnými slovami jazyka z hľadiska frekvencie použitia. To isté platí aj pre slová vo všeobecnosti, a slová, ktoré sú špecifické pre konkrétny predmet. Tento zákon platí aj pre iné aspekty ľudských jazykov. Platí to najmä pre frekvenciu výskytu N-gramov, a to ako inflexné formy a aj ako slovné zložky ktoré majú význam.



Obr. 12: Zipfianová distribúcia frekvencií N-gramov z technického dokumentu

Zo zipfovho zákona vychádza, že klasifikácia dokumentov pomocou frekvenčnej štatistiky N-gramov nebude veľmi citlivá na zníženie rozdelení na určitú hodnosť. To tiež znamená, že

ak porovnávame dokumenty z rovnakej kategórie mali by mať podobné frekvenčné rozdelenie N-gramov.

Z experimentov v [11] vyplýva, že pri trénovaní, vyťažené profily sa veľmi líšili medzi sebou v závislosti od svetového jazyka. V experimentoch použli korpus svetových jazykov a predmetom boli správy. Výsledkom je nasledujúca tabuľka.

Veľkosť textu	**≤ 300**	**≤ 300**	**≤ 300**	**≤ 300**	**> 300**	**> 300**	**> 300**	**> 300**
Veľkosť profilu	**100**	**200**	**300**	**400**	**100**	**200**	**300**	**400**
australia	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
brazil	70.0	80.0	90.0	90.0	91.3	91.3	95.6	95.7
britain	96.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0
canada	100.0	100.0	100.0	100.0	100.0	*99.6	100.0	100.0
celtic	100.0	100.0	100.0	100.0	99.7	100.0	100.0	100.0
france	90.0	95.0	100.0	*95.0	99.6	99.6	*99.2	99.6
germany	100.0	100.0	100.0	100.0	98.9	100.0	100.0	100.0
italy	88.2	100.0	100.0	100.0	91.6	99.3	99.6	100.0
latinamerica	91.3	95.7	*91.3	95.7	97.5	100.0	*99.5	*99.0
mexico	90.6	100.0	100.0	100.0	94.8	99.1	100.0	*99.5
netherlands	92.3	96.2	96.2	96.2	96.2	99.0	100.0	100.0
poland	93.3	93.3	100.0	100.0	100.0	100.0	100.0	100.0
portugual	100.0	100.0	100.0	100.0	86.8	97.6	100.0	100.0
span	81.5	96.3	100.0	100.0	90.7	98.9	98.9	99.45
Celkovo	**92.9**	**97.6**	**98.6**	**98.3**	**97.2**	**99.5**	**99.8**	**99.8**

*Označuje kombinácie testovacích premenných, ktoré boli horšie ako podobné kombinácie pomocou kratších profilov.

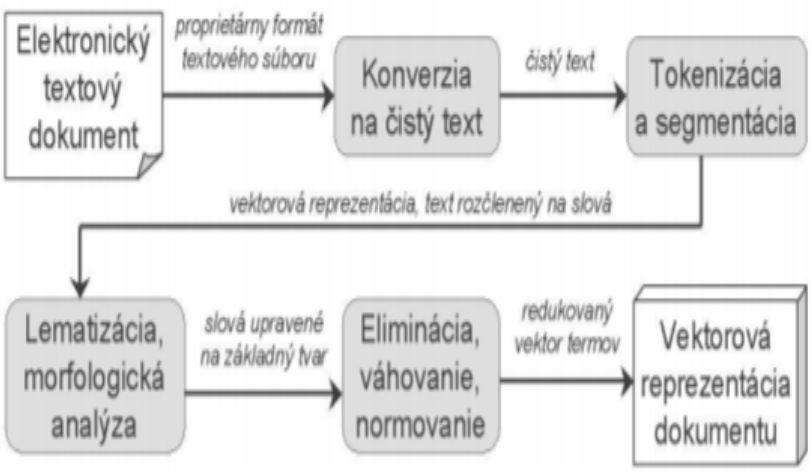
Konkrétny systém dosahuje úspešnosť klasifikácie 99.8% pri veľkosti profilov 300, 400 čo znamená že z 3478 textov iba 7x zlyhal. Takúto úspešnosť sa dá jednodnoznačne považovať za výbornú a spoľahlivú. Metódy založené na extrakcii N-gramov majú mnoho výhod medzi ktoré patria nízke nároky na pamäť, nenáročný proces trénovania, rýchla manipulácia s profilmi, odolnost voči gramatickým chybám.

5.2 Predspracovanie textu

Predpokladom pre aplikovanie metód na tvorbu profilov je transformácia textových dokumentov na reprezentačnú štruktúru vhodnú pre príslušné klasifikačné či zhlukovacie algoritmy. Tieto

sú najčastejšie založené na viacrozmernej analýze, ktorá na vstupe predpokladá údajovú m–rozmernú maticu pozorovaní na n objektoch. Objektmi sú v tomto prípade jednotlivé textové dokumenty v skúmanom korpuse. Pozorovania sú váhy jednotlivých kľúčových slov (resp. termov) v texte každého z dokumentov. Vyjadrujú hodnoty týchto charakteristických znakov, resp. premenných, pozorovaných na textových dokumentoch.[2]

Získanie príznakových popisov pre všetky textové dokumenty zo skúmaného korpusu sa súhrnne označuje ako predspracovanie textových údajov. Táto prípravná fáza aplikácie metód dolovania znalostí pozostáva zo sekvencie čiastkových krokov znázornených na obrázku nižšie.



Obr. 13: Schéma predspracovania textových údajov

Na vstupe sa predpokladajú textové dokumenty v rôznych formátoch. Prvým krokom je odstránenie redundantných formátovacích znakov a konverzia dokumentu na tzv. „čistý text“. Tento text sa ďalej delí na elementárne textové jednotky, tzv. tokeny. Následne sa v texte identifikujú slová, tzv. lexikálne jednotky, pre ktoré sa určí príslušný základný tvar (lema) a morfologické kategórie. Napokon sa odstránia neplnovýznamové slová, t.j. tie, pri ktorých sa predpokladá malý prínos k vyjadreniu celkového obsahu dokumentu. Zostávajúce plnovýznamové slová, ohodnotené vhodnou váhovou funkciou, potom tvoria hľadanú vektorovú reprezentáciu vstupného dokumentu. Pre zníženie výpočtovej náročnosti a zvýšenie efektívnosti algoritmov je potrebné, aby bol rozmer vektora príznakov čo najmenší. Na to sa používajú rôzne matematické metódy, kombinované s heuristickými metódami založenými na lingvistickej analýze textu. Tieto metódy spomedzi všetkých termov obsiahnutých v dokumente vyberajú tie, ktoré čo najlepšie charakterizujú daný dokument a čo najviac ho odlišujú od všetkých ostatných dokumentov.[1][2]

5.2.1 Konverzia vzstupného textu

Takáto konverzia je nutná ako prvý krok jej výsledkom je získaný čistý text. Takýto text sa dá definovať ako sekvenciu znakov a symbolov. Alfanumerické znaky nesú obshovú hodnotu

textu. Oddeľovače, interpunkčné znamienka členia textu. Vhodný čistý text z formátovaného dokumentu sa získa odstránením typografických značiek a netextových informácií (tabuľka, graf, obrázok, atď.)

Pri konverzí vstupného textu a ďalšej manipulácii s ním má zásadný význam jeho kódovanie. Štandardom a základným kódovaním pre angličtinu je ASCII (American Standard Code for Information Interchange). Obsahuje definície 128 znakov – 33 riadiacich znakov, 94 znakov pre tlač a znak medzery. ASCII znaky sú kódované na 7 bitoch. Kvôli spätnej kompatibilite s kódovaním ASCII a pod vplyvom niektorých ďalších problémov s praktickým používaním Unicode sa vytvorilo kódovanie UTF (Unicode Transformation Format) s premenlivou bitovou dĺžkou. UTF kóduje prvých 128 znakov zhodne s ASCII tabuľkou, čím sa zabezpečila spätná kompatibilita. Odlišujú sa až ďalšie znaky, ktoré sa kódujú viac ako ôsmimi bitmi. Azda najpoužívanejším je kódovanie UTF-8, ktoré je popísané v štandarde ISO 10646-1:2000.[2]

5.2.2 Členenie slov z textu

Segmentácia, členenie slov z textu je operácia pri ktorej sa v čistom vstupnom texte identifikujú jednotlivé textové slová. Sú to:

- reťazce alfanumerických znakov, oddelené medzerami alebo interpunkčními znamienkami
- jednotlivé interpunkčné znamienka

Príklad textu:

1. Context-based text analysis (diplomová práca)

Po rozčlenení sa rozdelí na textové jednotky:

[1][.] [Context] [-] [based] [text] [analysis] [(] [diplomová] [práca] [)]

V nasledujúcej fáze tokenizácie sa elementárne textové jednotky konvertujú na tzv. lexikálne jednotky – značky, resp. tokeny. Tokeny možno definovať ako systémom rozpoznané a akceptované skupiny znakov s kolektívnym významom, ktoré obyčajne zodpovedajú konkrétnemu slovníkovému záznamu. Identifikujú sa pomocou vopred zostavených slovníkov prípustných tvarov slov v kombinácii s rôznymi pravidlovými systémami.[2]

Pri tokenizácii sa postupnosti identifikovaných textových slov zhromažďujú tak, aby výsledkom bol vhodný tvar konrétneho slova. Napríklad textové jednotky [1] a [.] sa dajú zlúčit do tvaru [1.], ktorý zodpovedá radovej číslovke „prvý“.

5.2.3 Eliminovanie neplnovýznamových slov

Všetky identifikované a lematizované tokeny sú kandidátmi pre termy vektorového modelu textového dokumentu. Termy, čiže kľúčové slová, by mali čo najpresnejšie vyjadrovať obsah daného dokumentu. Zároveň je pre efektívnosť ďalšieho spracovania pomocou klasifikačných a zhlukovacích algoritmov výhodné minimalizovať počet termov popisujúcich jednotlivé dokumenty. Tieto

dve skutočnosti sú dôvodmi pre to, aby sa z ďalšieho spracovania vylúčili tokeny s malým príspevkom k celkovému obsahu textu. Takýmto slovám sa hovorí stop-slová.[2] Bývajú to hlavne neplnovýznamové slová – spojky, predložky, zámená, častice, rôzne netextové elementy a sčasti aj číslice a číslovky. Základné metódy na odstraňovanie stop-slov z textu sú v zásade dve. Prvý spôsob z tokenizovaného textu porovnáva slová s tzv. negatívnym slovníkom, na základe zhody sa následne odstránia slová z textu. Úspešnosť tejto metódy závisí od velkosti slovníka a jeho definície. Rovnako je nevýhodou aj že jeden slovník sa nedá použiť pre viacero jazykov.

Druhý spôsob je automatický, založený na odstraňovaní slov z textu ktoré sa v celom korpuse dokumentov vyskytujú s príliš veľkou (prípadne malou)frekvenciou. Podľa [12] majú dostatočne silnú rozlišovaciu schopnosť tie slová, ktorých frekvencia dokumentov patrí do intervalu $<\frac{N}{100}, \frac{N}{10}>$, kde N je počet dokumentov. Tento spôsob sa dá v praxi použiť pra viacero jazykov. Oproti prvému spôsobu ale dosahujú slabšie výsledky, ideálna je kombinácia oboch.

Pre angličtinu sú typické stop-slová:

a, about, above, after, again, an, and, any, are, be, before, behind, been, both, brief, can, come, did, didn't, down, during, each, else, et, etc, except, far, for, from, get, go, got, had, half, has, have, he, hello, his, how, in, into, it, just, last, let, low, me, most, much, my, near, no, none, not, now, of, off, on, our, out, own, past, per, rather, recent, say, see, self, she, so, soon, such, take, than, that, the, their, then, there, they, this, to, too, try, under, up, upon, us, use, via, want, was, we, were, what, when, who, why, would, yes, yet, ...

Pre slovenčinu sú typické stop-slová:

a, aby, aj, ako, ale, alebo, ani, áno, asi, bez, by, byť, cez, čo, či, dnes, do, ďalší, ešte, ho, i, iba, ja, je, jeho, jej, k, kam, každý, kde, kto, ktorý, ku, mať, môcť, môj, my, na, nad, nie, niet, než, nič, nový, o, od, on, po, pod, podľa, práve, prečo, pred, preto, potom, pri, prvý, s, sa, si, so, späť, svoj, tak, takže, teda, ten, tento, to, toto, tu, tuto, tvoj, ty, u, už, v, váš, viac, však, všetko, vy, z, za, že, ...

5.2.4 Lematizácia, stemming

V texte sa jednotlivé slová z pohľadu morfológie vyskytujú v rôznych tvaroch, je teda ich nutné previezt na tzv. lemy t.j. základny tvar. Lema[13] je základným, slovníkovým tvarom tokenu, čiže v tomto zmysle je totožná s lexikálnou jednotkou (napr. pekná-pekný, prípadov-prípad, atď.). Pri substantívach a adjektívach je to prvý pád jednotného čísla, pri slovesách neurčitok. Proces, ktorý z tvaru slova v texte určí základný tvar (najčastejšie odstránením slovotvorných, pádových a iných predpôn a prípon), sa nazýva lematizácia.

Špeciálnou formou lematizácie je izolácia koreňa slova (stemming), pri ktorej sa označkované slovo na danej pozícii nahrádza svojím kmeňovým základom, čiže sa konvertuje na základný tvar. Zo slov (tokenov) sa odstraňujú slovotvorné, pádové a iné predpony a prípony tak, že ostáva iba koreň slova, ktorý sa identifikuje ako term kľúčové slovo (term)[2]. Izolovaním koreňa slova sa výrazne zníži počet termov.

5.2.5 Lematizácia v angličtine

V angličtine, je izolácia koreňa pomerne jednoduchá, najpoužívanejším je Porterov algoritmus. Je založený na odstraňovaní prípon, pričom využíva pevný zoznam prípon a niektoré ďalšie pravidlá morfológie anglického jazyka. Vstupom pre Porterov algoritmus sú tokeny anglických slov, výstupom sú určené korene týchto slov.

Príklad Porterovho algoritmu:
Krok 1a - odstránenie prípon -s a -es:

SSES -> SS caresses -> caress
IES -> I ponies -> poni
S -> cats -> cat

Krok 1b - odstránenie prípon -d, -ed a -ing:

(m>0) EED -> EE agreed -> agree
*(*v*) ED -> plastered -> plaster*
*(*v*) ING -> motoring -> motor*

Krok 1c - zmena prípony -y na -i:

*(*v*) Y -> I happy -> happi*

Krok 2 - zmena prípon:

(m>0) ATIONAL -> ATE relational -> relate
(m>0) ENCI -> ENCE valenci -> valence

Krok 3 - zmena alebo odstránenie prípon:

(m>0) ICATE -> IC triplicate -> triplic
(m>0) ATIVE -> formative -> form
(m>0) ALIZE -> AL formalize -> formal

Krok 4 - odstránenie prípon:

(m>1) AL -> revival -> reviv
(m>1) ANCE -> allowance -> allow
(m>1) ENCE -> inference -> infer

Krok 5a - úprava koreňov, odstránenie -e:

(m>1) E -> probate -> probat
*(m=1 and not *o) E -> cease -> ceas*

Krok 5b - úprava koreňov, zmena -ll na -l:

*(m>1 and *d and *L) LL -> L controll -> control*

5.2.6 Lematizácia v slovenčine

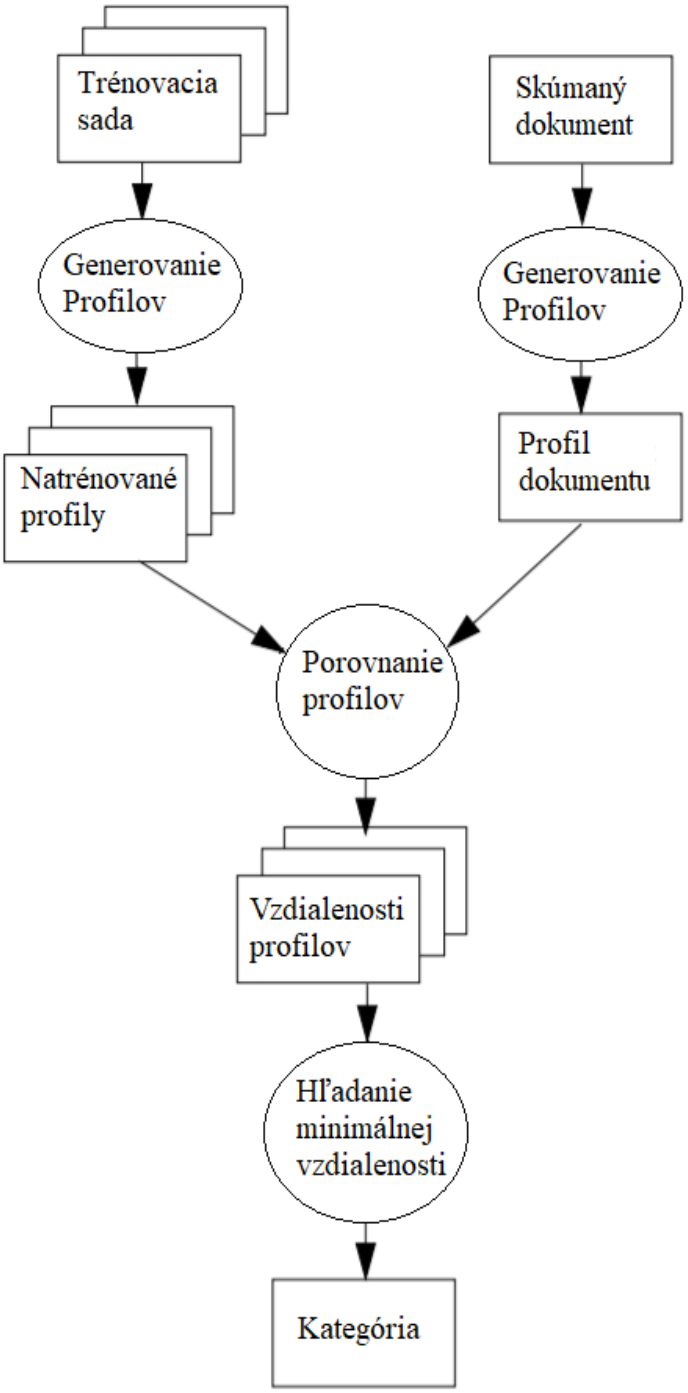
V slovenčine a podobne flektívnych jazykoch je lematizácia komplikovanejšia, pravidlá sú podstatne zložitejšie a izolácia koreňa sa rieši morfologickou analýzou a komplexným lingvistickým prístupom. Výsledok býva ale presnejší, na úkor zložitosti. Časté problémy sú:

- Tvarová homonymia:
  mier -> mieriť, ale aj mier, resp. miera
  mať -> mať (substantívum), ale aj mať (sloveso)
  veď -> viesť (sloveso), ale aj veď (častica), ...
- Lexikálna homonymia (jedno slovo označuje viac rôznych významov):
  oko: ľudské, morské, reťaze
  akcia: činnosť, podielový list
- Polysémia (viacznačnosť):
  padnúť: do studne, za vlasť, padol návrh, šaty jej padnú, ...

Nejednoznačnosti sa nedajú riešiť na úrovni morfológie (izoláciou koreňa), potrebná je komplexná jazyková analýza. Postup pri lematizácii v slovenčine:

- Slovám identifikovaným v texte počas tokenizácie sa priradí príslušný základný tvar – lema.
- Zároveň sa slová ohodnocujú príslušnými morfologickými kategóriami, na základe ktorých sa tokenom priradia gramatické značky – tagy. (tento proces sa tiež označuje ako tagovanie, angl. tagging)
- Značka určuje predovšetkým slovný druh, a potom, v závislosti od slovného druhu, aj ďalšie kategórie ako rod, číslo, pád, osobu, atď

5.3 Proces kategorizácie textu



Obr. 14: Proces kategorizácie textu

5.4 Porovnávanie profilov

Pre kategorizácií testovaného textu je klúčová operácia porovnávania profilov Pri tejto operácí sa porovnávajú všetky natrénované doménové profili s profilmi získaními zo skúmaného textu. Týmto spôsobom sa určí kategória ktorej profil je najviac zhodný s profilom skúmaného textu.

Navrhnutý proces porovnávania profilov je rozdelený na kroky. V provom kroku je skúmanému dokumentu vytvorený profil. Tento profil je vytvorený rovnakým spôsobom ako profily vytvorené pre v trénovacej sade. V ďalšom kroku sa profily porovnávajú na báze zvolených metód. V poslednom kroku sa nájdením napodobnejších profilov zvolí výstupná kategória.

5.5 Metódy pre porovnávanie profilov

Existuje viacero porovnávacích metód ktoré určujú podobnosť profilov. V tejto práci sa venujem dvom vybraným metódam a to na základe početnosti a požicie N-gramu a v druhom prípade na základe váženej početnosti. Ďalej sa v práci venujem ich porovnaniu a vyhodnoteniu úspešnoti kategorizácie v závislosti od rôznych parametrov.

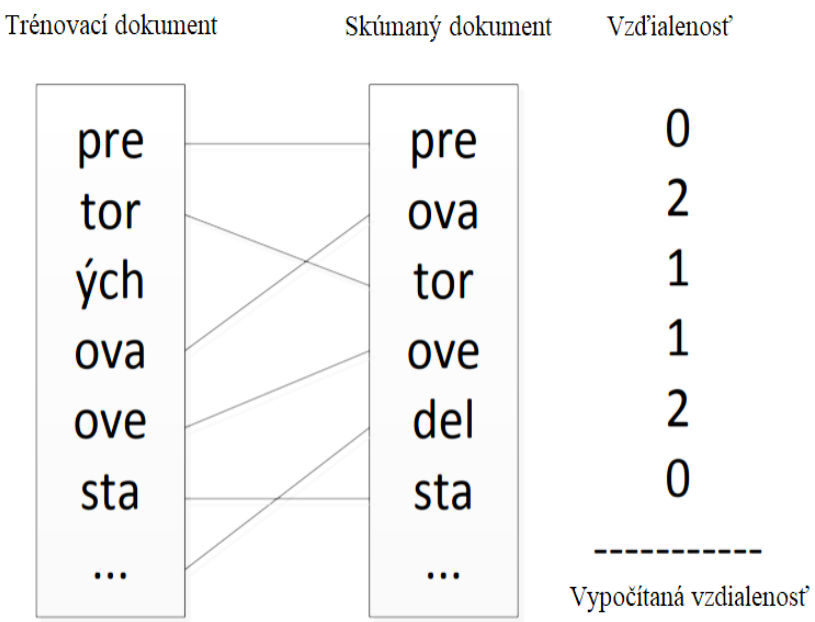
5.5.1 Porovnávanie profilov na základe vzdialeností a pozície N-gramu

Táto metóda porovnáva vygenerované profily na základe vypočítaných vzdialeností a rozmiestnením N-gramov v zoradenom profile. Metóda pracuje tak že vytvorí trénovacím a skúmaným N-gramom profily. Rozdielnosť týchto profilov predstavuje súčet rozdielov polohy N-gramov zo skúmaného textu od natrénovaných N-gramov.

$$X = \sum_{n=1}^{N} abs(A_n - B_n)$$

kde X je pre danú kategóriu hodnota rozdielnosti skúmaných profilov, A_n a B_n vyjadrujú umiestnenie ntého N-gramu v skúmanom a porovnávanom profile.

Z tohto ďalej vyplýva, že pri katogorizácií N trénovacích profilov je nutné vykonať tento výpočet s každým takýmto profilom - N krát. Výsledná kategória sa určí výberom najmenšej hodnoty z vypočítaných hodnôt X.



Obr. 15: Meranie vzdialenosti

5.5.2 Porovnávanie profilov na základe váženej početnosti

Rozdiel oproti predošlej metóde je že sa neporovnáva pozícia N-gramov ale berie sa do úvahy vážená početnosť. Túto metódu opisuje nasledujúci vzťah:

$$Y = \frac{\sum_{a=1}^{A} P_n}{\sum_{b=1}^{B} P_b}$$

Y je vypočítaná hodnota váženej početnosti, čitateľ predstavuje súčet početností v porovnávaných N-gramov zo skúmaného profilu. Menovateľ predstavuje sumu výskytov N-gramov nachádzajúcich sa v doménovom profile. Výpočet sa musí vykonať nad každým trénovacím profilom. Výslednú kategóriu určí maximálna hodnota váženej počtnosti s vypočítaných hodnôt Y.

6 Experimenty

Aby sme mohli hodnotiť presnosť klasifikácie, musíme nájsť vhodné hodnotiace metriky. Najjednoduchšia a napriek tomu často používana metrika, je percentuálna úspešnosť. Táto metrika je využívaná aj v experimentoch v tejto práci. Je daná vzťahom:

$$P = \frac{C}{I} * 100$$

P predstavuje presnosť klasifikácie, C správne klasifikovaný text, I nesprávne klasifikovaný text. Výsledok je udávaný v percentách.

6.1 Testovanie a vyhodnotenie

Postup testovania je nasledovný pre zvolené kategórie šport, politika, veda:

1. Vyberie sa testovaná kategória z už natrénovaných dokumentov ktorá bude následne hľadaná v skúmanom texte.
2. Zadajú sa parametre testu ako sú dĺžka textu, počet dokumentov.
3. Vyhodnotenie testu. Výsledok obsahuje percentuálnu úspešnosť testu a teda zhodu s kategóriou.

6.1.1 Testovanie úspešnosti kategorizácie na základe jazyku

Tento test bol zameraný na porovnanie úspešnosti kategorizácie vzhľadom na použitý jazyk v dokumentoch. Testovaná vzorka mala veľkosť 1000 textov pre každý jazyk. Výsledky uvádzam v nasledujúcich tabuľkách. O niečo lepšiu úspešnosť vykazovala v mojich testoch kategorizácia slovenčiny. Najslabšie výsledky pri oboch jazykoch predstavovala kategória politika. Výsledky testu uvádzam v nasledovných tabuľkách.

Test	1	2	3	4	5	6	7	8	9	10
Úspešnosť SK	75	77	83	80	79	85	79	82	81	80
Úspešnosť EN	80	60	65	70	78	78	75	70	80	79

Tabuľka 3: Test kategórie šport

Test	1	2	3	4	5	6	7	8	9	10
Úspešnosť SK	75	78	80	82	74	74	75	81	79	76
Úspešnosť EN	90	60	62	80	69	70	71	73	78	76

Tabuľka 4: Test kategórie politika

Test	1	2	3	4	5	6	7	8	9	10
Úspešnosť SK	83	80	84	81	76	82	90	92	85	86
Úspešnosť EN	80	85	79	78	79	80	88	88	89	70

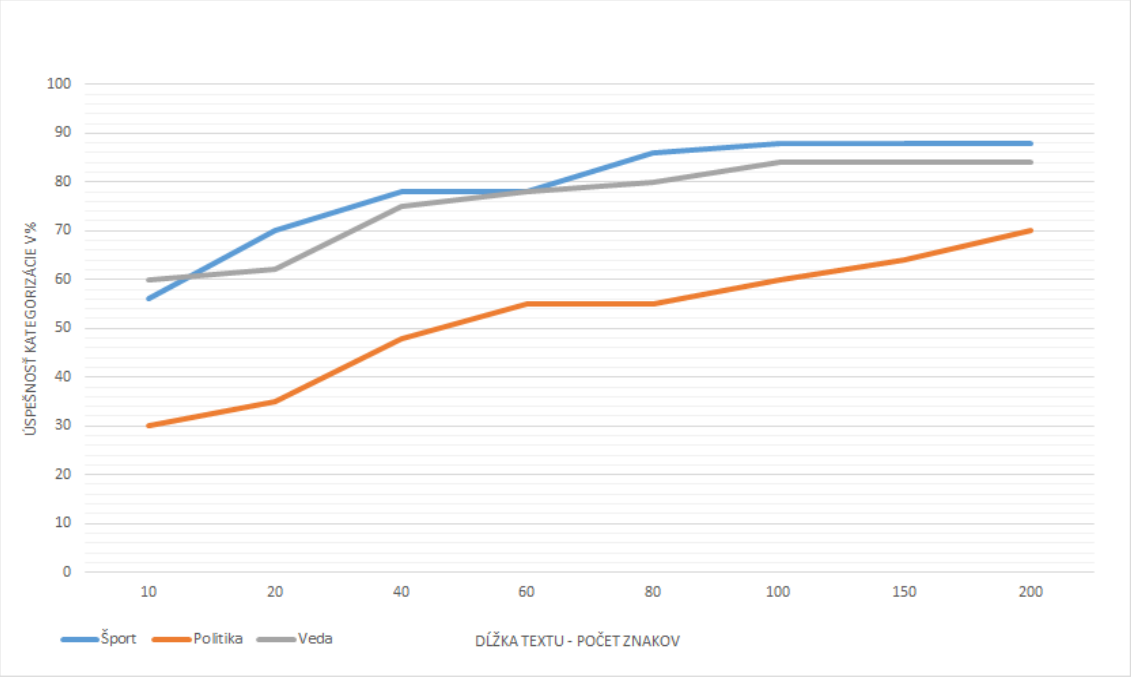
Tabuľka 5: Test kategórie veda

6.1.2 Testovanie úspešnosti kategorizácie na základe dĺžky textu

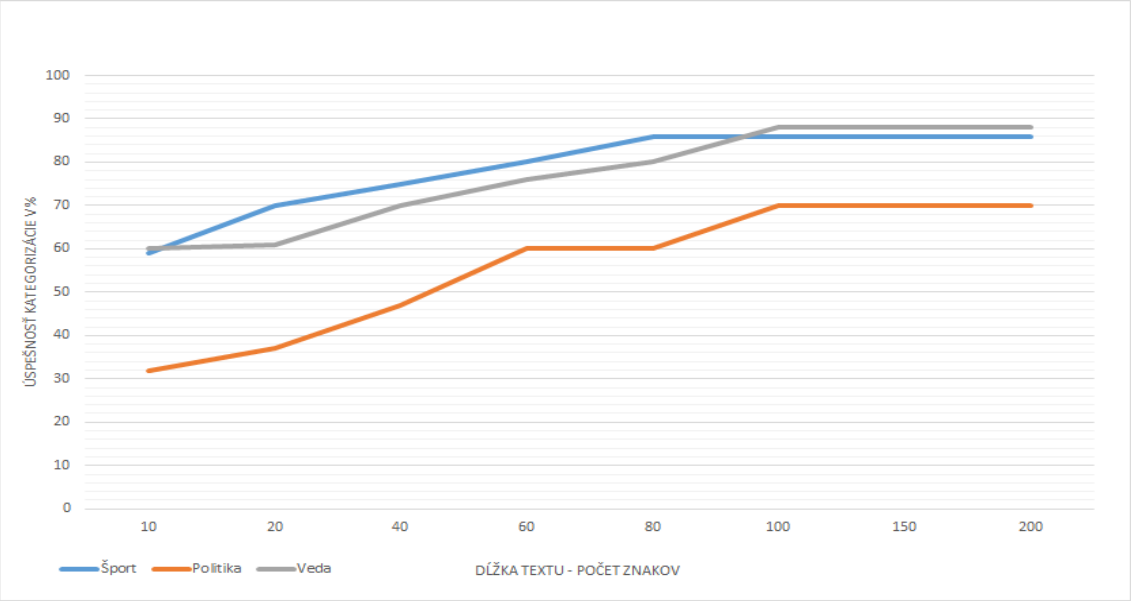
V tomte teste som sa venoval závislosti úspešnosti kategorizácie od dĺžky skúmaného textu. Rovnako ako aj v predchádzajucom teste som testy vykonal na všetkých kategóriach pre oba jazyky. Veľkosť znakov sa pohybovala od 10 po približne 200 znakov. Testovaná vzorka mala veľkosť 1000 textov pre každý jazyk. Z výsledkov badať markantný rozdiel v testovaných kategóriach kde pri vede a športe sa dosahahuje omnoho lepšia úspešnoť ako v kategórii politika.

6.1.3 Výsledky pre metódu vzdialenosti a pozície N-gramov

Pri oboch zvolených jazykoch je badať trend, že od určitej dĺžky textu úspešnoť metódy ďalej nerastie. Z toho vyplýva predpoklad že táto metóda sa viac hodí pre kratšie texty.



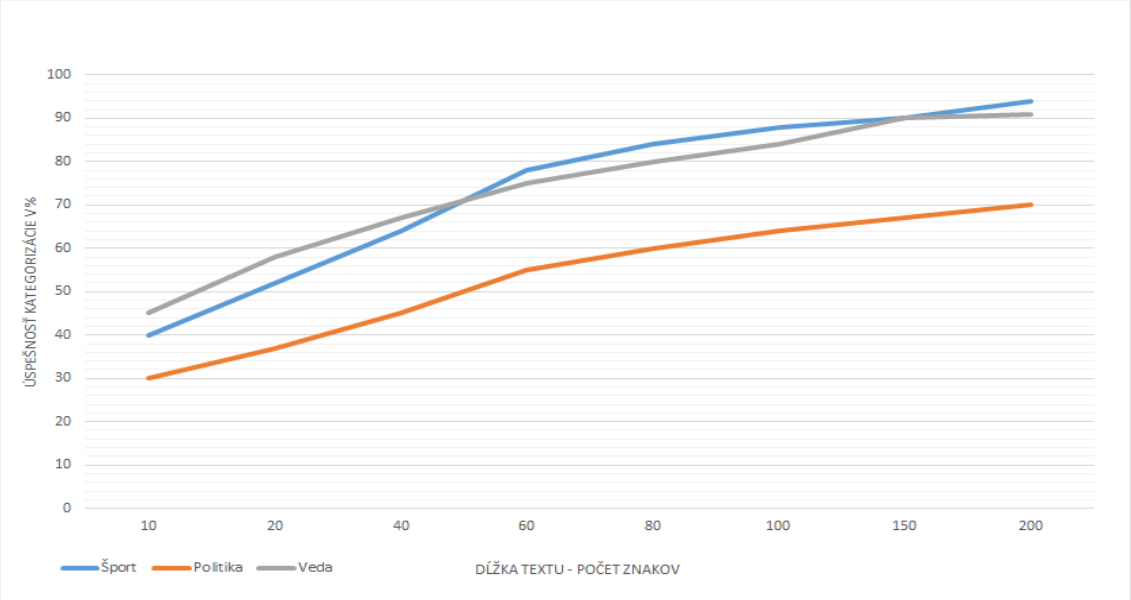
Obr. 16: Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Slovenský jazyk



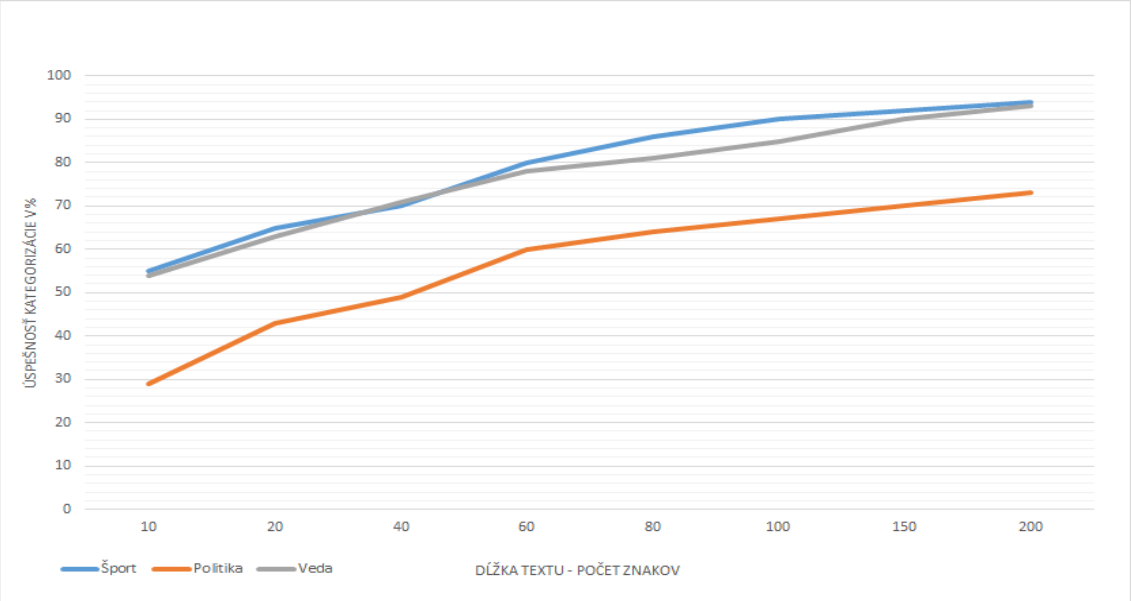
Obr. 17: Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Anglický jazyk

6.1.4 Výsledky pre metódu váženej početnosti

Pri zvolených parametroch testu dosahovala táto metóda podpriemerné výsledky pri krátkych textoch. Rast úspešnosti je ale naproti prechádzajúcej metóde lineárnejší a je teda možné predpokladať že s narastajúcou dĺžkou textu bude úspešnosť ďalej rásť.



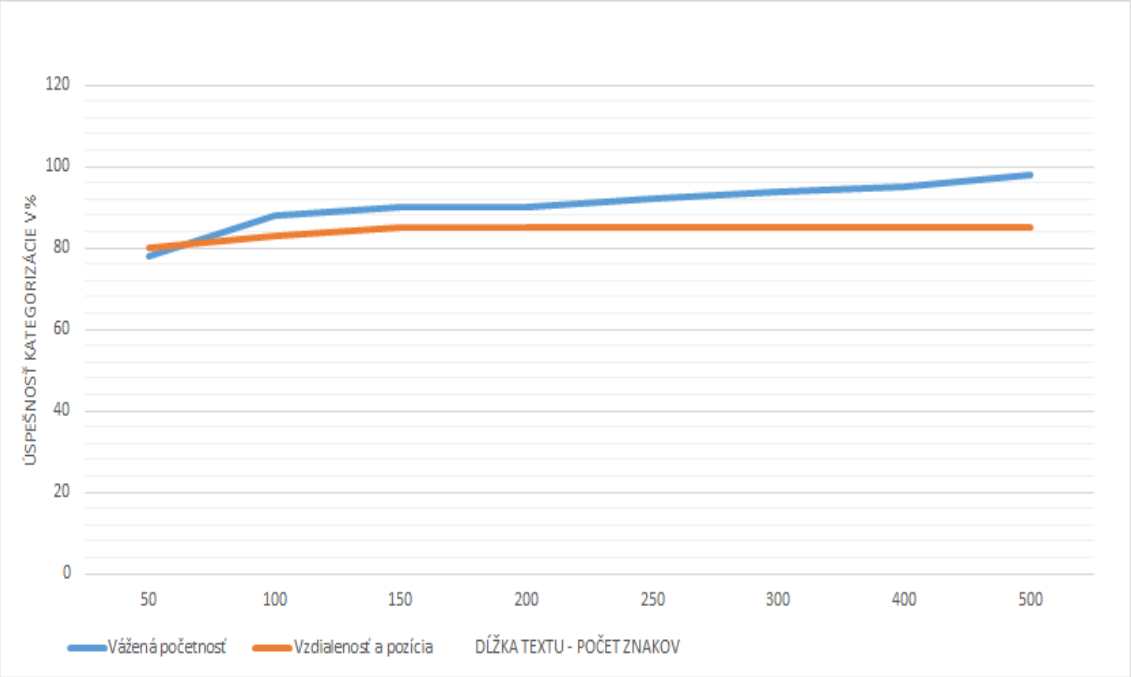
Obr. 18: Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Slovenský jazyk



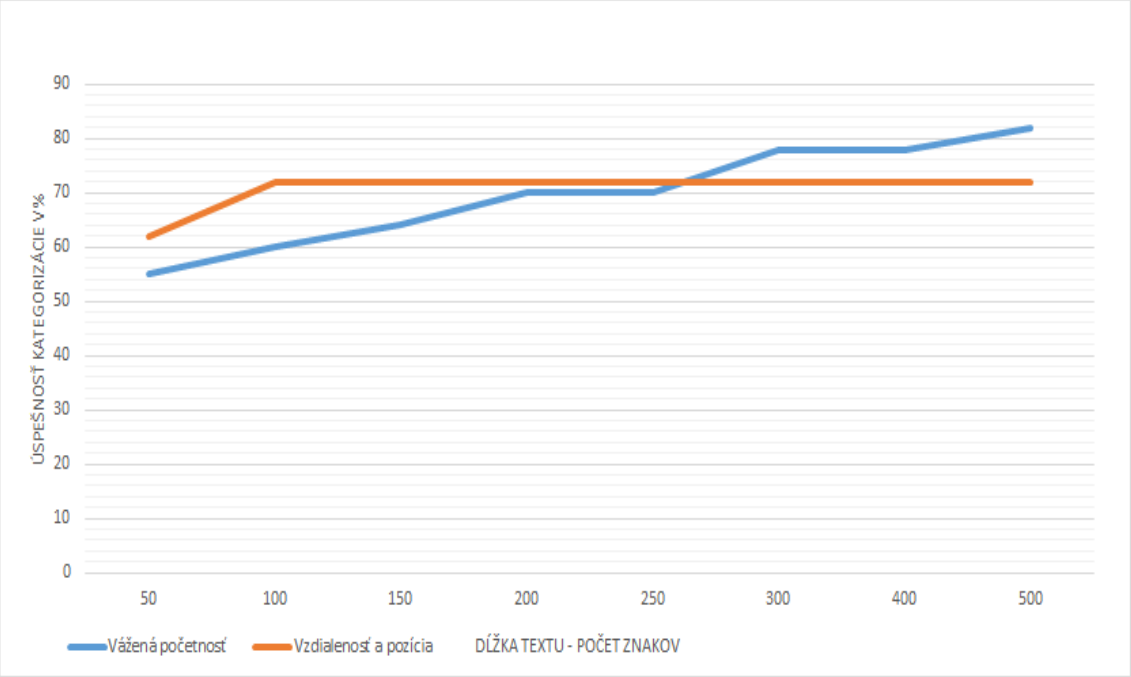
Obr. 19: Závislosť úspešnosti kategorizácie od dĺžky textu. Jazyk: Anglický jazyk

6.2 Porovnanie zvolených metód

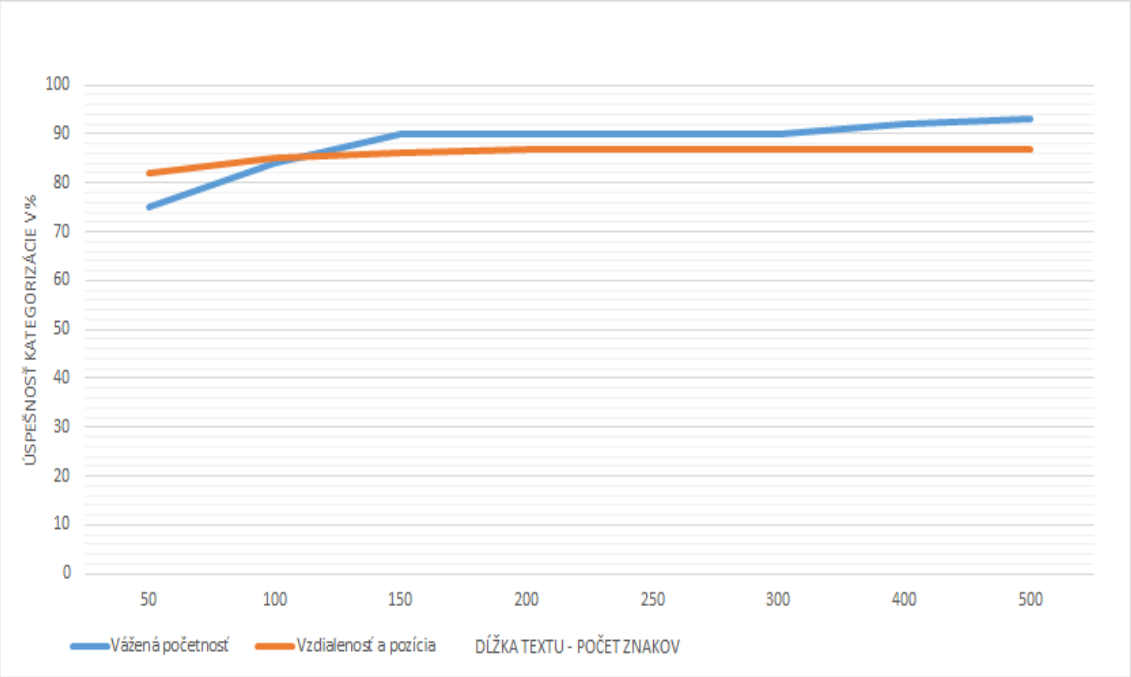
Cieľom v tomto teste bolo vykonať porovnanie dvoch zvolených metód a určit odlišnosti. Keďže z predchádzajúceho testu je badať rozdielnosť metód na základe dĺžky testovaného textu, rozhodol som sa ich porovnať ešte raz s upravenými parametrami. Test prebehol na rovnakej vzorke dát pre obe metódy a pre všetky natrénované kategórie. Veľkosť znakov sa pohybovala od 50 po približne 500 znakov. Testovaná vzorka mala veľkosť 2000 textov. Z výsledkov sa teda potvrdilo že metóda ktorá funguje na báze pozície N-gramov fungovala lepšie pri kratších textoch a všeobecne mala vo všetkých testoch vyrovnanú úspešnosť. Druhá metóda pracujúca na základe váženej početnosti dosahovala výborne výsledky pri dlhších textoch.



Obr. 20: Porovnanie zvolených metód. Kategória: šport



Obr. 21: Porovnanie zvolených metód. Kategória: politika



Obr. 22: Porovnanie zvolených metód. Kategória: veda

7 Záver

Diplomová práca sa zaoberá kontextovou analýzou textu. V práci sa venujem analýze metód pre kategorizáciu textu slovenského a anglického jazyka. Cieľom práce bolo implementovať zvolené metódy a na experimentoch dokázať a porovnať ich funkčnosť. Práca pozostáva zo siedmich kapitol a popisuje problematiku analýzy textu od dolovania až po proces samotnej kategorizácie. V prvých dvoch kapitolách opisujem súčastný stav problematiky kde preberám aktuálne aj historické spôsoby analýzy textových dát. Ďalej preberám zásady predspracovania dát. V tretej kapitole opisujem konkrétne dáta použité v tejto práci. Definujem korpus pre slovenský a anglický jazyk. Vo zvyšných kapitolách sa venujem špecifickým metódam vhodných pre kategorizáciu textu. Opisujem celý proces so zvolenými metódami ktoré následne na testovacích dokumentoch otestujem. Veľký dôraz venujem experimentom kde demonštrujem výsledky práce.

Literatúra

[1] Aggarwal, C. C., & Zhai, C. (Eds.). (2012). *Mining text data.* Springer Science & Business Media.

[2] Paralič, J., Furdík, K., Tutoky, G., Bednár, P., Sarnovský, M., Butka, P., & Babič, F. (2010). *Dolovanie znalostí z textov.* Equilibria, Košice.

[3] Páleš, E. (1994). Sapfo. *Parafrázovač slovenčiny. Bratislava: Veda.*

[4] Fürnkranz, J. (1998). A study using n-gram features for text categorization. *Austrian Research Institute for Artifical Intelligence*, 3(1998), 1-10.

[5] Kennington, Casey R. (2011). *Application of Suffix Trees as an Implementation Technique for Varied-Length N-gram Language Models.* Master's thesis. Universität des Sarlandes. 57 s.

[6] McEnery, T., & Wilson, A. (2003). Corpus linguistics. *The Oxford handbook of computational linguistics*, 448-463.

[7] Češka, Z., Hanák, I., & Tesař, R. (2008). Extrakce N-gramů z rozsáhlých textů. In *Konference Znalosti.*

[8] Kit, C., & Wilks, Y. (1998, November). The Virtual Corpus approach to deriving n-gram statistics from large scale corpora. In *Proceedings of 1998 International Conference on Chinese Information Processing* (pp. 223-229).

[9] Tesař, R. (2007) The Use of N-Grams in Text Categorization. *Západočeská univerzita v Plzni.* Disertační práce

[10] Abouelhoda, M. I., Kurtz, S., & Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays. *Journal of discrete algorithms*, 2(1), 53-86.

[11] Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. *Ann arbor mi*, 48113(2), 161-175.

[12] Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613-620.

[13] ŠÍMA, Jiří and Roman NERUDA. *Teoretické otázky neuronových sítí.* Vyd. 1. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9.

[14] Garabík, R., Gianitsová, L., Horák, A., Šimková, M. (2004). Tokenizácia, lematizácia a morfologická anotácia Slovenského národného korpusu. (Current version of May 4, 2004) *SNK JÚĽŠ*, Bratislava.

[15] Pappu, A. (2009). Using wikipedia for hierarchical finer categorization of named entities. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, Volume 2 (Vol. 2).

[16] Soori, H., Prilepok, M., Platos, J., & Snášel, V. (2015). Utilizing text similarity measurement for data compression to detect plagiarism in Czech. In *Afro-European Conference for Industrial Advancement* (pp. 163-172). Springer, Cham.

[17] Slovenský národný korpus, Jazykovedný ústav Ľ. Štúra Slovenskej akadémie vied `http://www.korpus.sk`, Navštívené 7.3.2018